# DEVELOPMENT and PERFORMANCE EVALUATION of an APPLICATION for NEWS ARTICLE SUMMARIZATION, CLASSIFICATION, and SENTIMENT ANALYSIS using DEEP LEARNING MODELS

**A project report submitted in partial fulfilment of the requirement for**

**the award of the degree of**

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE & ENGINEERING**

of

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR**

by

| | |
|---|---|
| **JEERA AMBIKA** | **212K1A0515** |
| **S B HIMACHANDRI** | **212K1A0547** |
| **SHAMSHUDDIN M G M** | **212K1A0552** |
| **SAKARAY VENKATESH** | **212K1A0544** |
| **NALBANDA SHAIK SHAVALI** | **212K1A0536** |

**Under the Esteemed Guidance of**

**K. ARJUN** M. Tech., (Ph. D.)
**Associate Professor**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BHEEMA INSTITUTE OF TECHNOLOGY AND SCIENCE**

**ALUR ROAD, ADONI – 518301**
**KURNOOL, ANDHRA PRADESH.**
**2021- 2025**

**BHEEMA INSTITUTE OF TECHNOLOGY AND SCIENCE**

**ALUR ROAD, ADONI-518301, KURNOOL, A.P.**

(Affiliated to Jawaharlal Nehru Technological University Anantapur, Anantapuramu)

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## CERTIFICATE

This is to certify that the project entitled **" DEVELOPMENT and PERFORMANCE EVALUATION of an APPLICATION for NEWS ARTICLE SUMMARIZATION, CLASSIFICATION, and SENTIMENT ANALYSIS using DEEP LEARNING MODELS"** being submitted by **Ms. J.Ambika (212K1A0515), Ms. SB.Himachandri (212K1A0547), Mr. Shamshuddin M G M (212K1A0552), Mr.Sakaray Venkatesh (212K1A0544) & Mr.N.Shaik Shavali (212K1A0536)** in partial fulfilment of the requirement for the Degree of **Bachelor of Technology in Computer Science & Engineering of JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR, Anantapuramu** during the year **2021 - 2025**.

| | |
|---|---|
| K. ARJUN | Dr. D. William Albert |
| **Project Guide** | **Head of Department** |

Place: ADONI

Date:

Certify that the candidate was examined by me in the Viva Voice Examination held at Bheema Institute of Technology and Science, Alur Road, Adoni on

________________.

| | |
|---|---|
| **INTERNAL EXAMINER** | **EXTERNAL EXAMINER** |

# GUIDE DECLARATION

This is to certify that the project entitled **"DEVELOPMENT and PERFORMANCE EVALUATION of an APPLICATION for NEWS ARTICLE SUMMARIZATION, CLASSIFICATION, and SENTIMENT ANALYSIS using DEEP LEARNING MODELS"** done by **Ms.J.Ambika (212K1A0515), Ms.SB.Himachandri (212K1A0547), Mr.Shamshuddin M G M (212K1A0552), Mr.S.Venkatesh (212K1A0544) & Mr.N.Shaik Shavali (212K1A0536)** has been conducted under my direct supervision and guidance.

This study is in partial fulfilment for the award of **BACHELOR OF TECHNOLOGY** from **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR**, Anantapur, and Andhra Pradesh. Further it is certified that project has not been previously submitted to any other university for the requirement of **BACHELOR OF TECHNOLOGY**.

**Date:** **Mr. K ARJUN**

**Place**: **ADONI** **Associate Professor**
**Dept of CSE.**

# STUDENT DECLARATION

We, **Ms.J.Ambika (212K1A0515), Ms.SB.Himachandri (212K1A0547), Mr.Shamshuddin M G M (212K1A0552), Mr.S.Venkatesh (212K1A0544) & Mr.N.Shaik Shavali (212K1A0536)** student of **Bheema Institute of Technology and Science, Adoni**, hereby declare that the dissertation entitled **"DEVELOPMENT and PERFORMANCE EVALUATION of an APPLICATION for NEWS ARTICLE SUMMARIZATION, CLASSIFICATION, and SENTIMENT ANALYSIS using DEEP LEARNING MODELS"** embodies the report of my project work carried out independently by me during final year of **Bachelor of Technology in Computer Science & Engineering** under the supervision and guidance of **K. ARJUN M. Tech., (Ph. D.) Associate Professor**, **Department of Computer Science & Engineering**, **Bheema Institute of Technology and Science, Adoni, A.P.,** and this work has been submitted for the partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology**.

We have not submitted the matter embodies to any other University or Institutions for the award of any other degree.

| STUDENT NAME | REGISTER NUMBER |
|---|---|
| J AMBIKA | 212K1A0515 |
| SB HIMACHANDRI | 212K1A0547 |
| SHAMSHUDDIN M G M | 212K1A0552 |
| S VENKATESH | 212K1A0544 |
| N SHAIK SHAVALI | 212K1A0536 |

# ACKNOWLEDGEMENT

We take this opportunity to thank all those magnanimous persons who rendered their full support to our work the pleasure, the achievement, the glory, the satisfaction, the reward, appreciation and the construction of this regular schedule spared their valuable time for us. They have been guiding and source of inspiration towards the completion of this project.

We are very grateful to **Sri P. N. VISHNU VARDHAN REDDY**, Secretary of Bheema Institute of Technology and Science, for providing us with a wonderful learning environment.

We express our gratitude to **Sri G. S. SURENDRA BABU**, principal of Bheema Institute of Technology and Science, Adoni, for giving us the opportunity to undertake this project.

We also like to express our sincere gratitude to **Sri Dr. D. William Albert**, professor and Head of the Department of Computer Science and Engineering, for his continuous guidance and unwavering support towards the completion of this project.

We are thankful to our project guide, **Sri K. ARJUN M. Tech., (Ph. D.)**, Associate professor, who with his continuous efforts, unfailing interest, constant support and providing me the right infrastructure helped me in completing this project work.

We also extend our thanks to all the faculty members of CSE department and our friends for their valuable suggestions and support which directly or indirectly contributed to shaping this project into a comprehensive one.

Finally, we express our deepest gratitude to our **parents**, whose unconditional love, support, and encouragement have been a pillar of strength throughout our education.

| **STUDENT NAME** | **REGISTER NUMBER** |
|---|---|
| **J AMBIKA** | **212K1A0515** |
| **S B HIMACHANDRI** | **212K1A0547** |
| **SHAMSHUDDIN M G M** | **212K1A0552** |
| **S VENKATESH** | **212K1A0544** |
| **N SHAIK SHAVALI** | **212K1A0536** |

# ABSTRACT

With the exponential growth of online news articles, it has become increasingly difficult for users to consume vast amounts of information efficiently. This project, **"Development and Performance Evaluation of an Application for News Article Summarization, Classification, and Sentiment Analysis using Deep Learning Models,"** leverages **Natural Language Processing (NLP)** techniques to automate the process of summarizing and categorizing news articles. Using **DistilBERT**, a state-of-the-art transformer model from **Hugging Face**, this system processes news content and generates concise yet informative summaries. Additionally, it classifies articles into relevant categories, helping users quickly identify key topics. The dataset used for this project is **"News Article (Weekly Updated)"** from **Kaggle**, provided by **The Star Malaysia**, a reputable news source. This dataset ensures a diverse range of news coverage. The implementation is built using **Python** and various libraries, including **Streamlit** for the web interface, **Pandas** for data handling, **NLTK** for text pre-processing, and **Transformers & Torch** for model operations. The system's performance is evaluated using **ROUGE Score** for summarization accuracy and **classification accuracy metrics** for topic prediction. This project addresses the challenge of **information overload** by providing a streamlined way to consume news efficiently. Future enhancements could involve multi-lingual support and real-time article updates.

# CONTENTS

# Chapter 1: INTRODUCTION

In an era where digital content is exploding at an unprecedented rate, the need to efficiently process and interpret vast volumes of news articles has become increasingly important. This project, titled **"Development and Performance Evaluation of an Application for News Article Summarization, Classification, and Sentiment Analysis using Deep Learning Models"** employs state-of-the-art Natural Language Processing (NLP) techniques to condense and categorize news content rapidly and accurately. This chapter introduces the project by providing essential background information, outlining the challenges in handling extensive textual data, and detailing the objectives that drive this research and implementation.

## 1.1: BACKGROUND & MOTIVATION

The digital age has revolutionized the way information is disseminated, with news outlets and digital media platforms generating large volumes of textual content daily. With the proliferation of online news, stakeholders such as researchers, educators, professionals, and decision-makers are often challenged by the overwhelming amount of unstructured data available. News article summarization and classification serve two critical roles in addressing this challenge:

- **Summarization:** With continuous streams of news, obtaining succinct overviews becomes crucial for time-sensitive decision-making. Summarized information helps readers grasp the main points of an article without having to navigate through redundant or superfluous content.
- **Classification:** Assigning predefined categories to articles enhances the ability to filter and retrieve pertinent information. Effective classification can streamline analytical processes, supporting trends analysis, sentiment analysis, and real-time monitoring of events.
- Traditional methods struggle to balance the need for conciseness and completeness when summarizing text or the accuracy when classifying diverse news topics. Advancements in NLP, particularly in transformer-based models like **DistilBERT**, a distilled version of the **BERT** [**Bidirectional Encoder Representations from Transformers**] model, offers the advantages of enhanced computational speed and reduced resource demands without

significantly sacrificing accuracy making it an ideal candidate for this project.

**THE ROLE OF NLP IN MODERN INFORMATION PROCESSING:**

The integration of NLP in processing news articles represents a significant leap forward in information management. When examining extensive news datasets like THESTAR.COM.MY from Kaggle, automated processing is not only beneficial but often necessary to produce reliable insights. NLP techniques facilitate several key functionalities:

1. **Text Summarization:** Automated summarization techniques reduce the workload involved in manually curating content, enabling quick extraction of key points from lengthy articles. This capability is particularly relevant in scenarios like monitoring breaking news or augmenting alert systems in emergency management.
2. **Text Classification:** With classification, articles are mapped to specific topics or categories. This task involves understanding the context, tone, and semantic structure of the text, which traditional keyword-based methods often fail to capture adequately. By leveraging transformer models, the project seeks to address these shortcomings, achieving higher accuracy and adaptability.

## 1.2 OBJECTIVE OF THE PROJECT

The primary goal of this project is to develop a comprehensive system that leverages NLP techniques for news article summarization, classification and sentiment analysis to evaluate the model's performance in resource-constrained environments. The system is built with several targeted functionalities and performance metrics in mind:

- **Application Development**: Build an end-to-end application for news article summarization, classification and for performing sentiment analysis using DistilBERT.
- **Implementation of a Pre-trained DistilBERT Model:** Integrate a fine-tuned version of DistilBERT from Hugging Face's model repository to enhance both summarization and classification capabilities. Thus, evaluate whether the model is efficient in delivering results with reduced computational overhead.
- **Utilization of the THESTAR.COM.MY Dataset:** Utilize domain-specific corpora such as "News Article (Weekly Updated)" from Kaggle, provided by The Star Malaysia.

- **Development of a User-Friendly Web Interface:** Create an interactive front-end platform using Streamlit that enables users to input news articles, view summaries, observe classification and check sentiment analysis outputs seamlessly. This web interface is intended to bridge the gap between the underlying complex NLP algorithms and end-users who benefit from simplified visualizations and outputs.
- **Performance Benchmarking:**
  - **Evaluation**: Rigorously evaluate the model's performance in resource-constrained environments and its ability to process large news datasets.
    - **Metrics**: Accuracy, inference time (ms), and memory usage (MB).
- **Real-World Testing:** Deploy the application on low-resource devices (e.g., laptops, edge devices) and collect user feedback on responsiveness and usability.

### Components and Technical Flow

The technical architecture of the project is designed to ensure efficiency and scalability. The project employs a range of modern tools and libraries, which are detailed as follows:

- **Python as the Core Language:** The choice of Python is extensive ecosystem and wide acceptance in the data science community. Libraries like Pandas and NumPy are used in handling data preprocessing, while Requests and BeautifulSoup4 facilitate the extraction of textual data from online resources.
- **Streamlit for the Web Interface:** Streamlit is chosen to develop the web UI because of its simplicity and ease-of-use, which allows for rapid prototyping and deployment of interactive applications. Its native support for integrating numerous Python libraries significantly aids in visualization of outputs, as summarized text and classification results.
- **Integration of NLP Tools and Frameworks:** The project integrates key libraries like Transformers and Torch to harness the power of pre-trained language models. Additionally, NLTK is utilized for various text processing tasks, and Matplotlib along with WordCloud serves for the visual representation and analysis of text frequency and distribution patterns.
- **Data Management and Output Visualization:** For data wrangling and output storage, libraries like Openpyxl are deployed to manage Excel files containing summarized and classified data. Summarizer and WordCloud provide innovative

approaches to text summarization and visualization, while Matplotlib offers detailed graphical representations.

- **System Requirements for Optimal Performance:** Recognizing the potential computational load, the project is designed to operate on mid-range hardware specifications. This focus on moderate system requirements ensures wider accessibility and practical deployment in various resource-constrained environments.

**Applications and Future Relevance**

The impact of this project extends far beyond academic exercises; it is designed to provide tangible benefits with broad applications:

- **Real-Time News Monitoring and Alerts:** In fast-paced scenarios such as stock market analysis, emergency response, and political events, real-time summarization can offer critical insights at the moment they are needed. The classification component further refines this by categorizing news items into relevant themes.
- **Academic and Organizational Research:** For researchers and students delving into media studies, journalism, and communications, an automated system for summarizing and classifying news articles offers a valuable tool that reduces manual analysis time while increasing the reliability of the insights drawn from large datasets.
- **Customized User Experiences:** The system’s design also lays the groundwork for personalized content delivery. By understanding individual user preferences and common biases in news consumption, future iterations of the system could offer tailored news feeds that cater to the specific interests and needs of diverse users.
- **Enhanced Data Analytics and AI-driven Insights:** With the continued advancements in AI and data analytics, this project illustrates a critical step toward integrating automated understanding of textual data into broader analytical frameworks. This integration facilitates the transition from simple data collection to intelligent data interpretation, creating a foundation for advanced decision-making systems.

### Bridging Research and Practical Implementation

While the academic community has explored various high-level theories in NLP, real-world implementation often presents challenges that require an intricate balance of theoretical knowledge and practical considerations. By utilizing a pre-trained DistilBERT model, this project bridges the gap between cutting-edge research and pragmatic system development. The careful selection of libraries and the adoption of a scalable framework ensures that our approach is not only robust in theoretical insights but also operationally efficient.

The integration of evaluation metrics such as the ROUGE score and classification accuracy further reflects the project's commitment to measurable and reproducible outcomes. These metrics provide clear benchmarks that can be used to compare the performance of our system with existing methodologies, establishing a concrete basis for iterative improvements.

### Impact on the Field of Natural Language Processing

This project showcases the capabilities of modern NLP frameworks in transforming how we process and consume vast amounts of information. By automating summarization and classification, it democratizes access to streamlined, high-quality news content for academia, media organizations, and the broader public. In an age of information overload, such systems pave the way for more efficient, transparent, and responsive information ecosystems.

Moreover, the methodologies and techniques documented in this project serve as a valuable reference for future research and development in textual data analysis. By detailing every step—from dataset and model selection to performance evaluation—we provide a comprehensive resource to inspire further innovations.

This project reflects the broader movement in computer science and artificial intelligence towards more intuitive, accessible, and intelligent systems. The foundational principles and lessons learned from "Development and Performance Evaluation of an Application for News Article Summarization, Classification, and Sentiment Analysis using Deep Learning Models" will continue to influence future advancements in automated text processing.

# Chapter 2: LITERATURE SURVEY

The evolution of Natural Language Processing (NLP) over the past decade has transformed how textual data is processed, understood, and utilized. This chapter reviews previous research, methodologies, and state-of-the-art techniques in news article summarization and classification, with a focus on transformer-based models.

The goal is to position our project - "**Development and Performance Evaluation of an Application for News Article Summarization, Classification, and Sentiment Analysis using Deep Learning Models**" - within the broader research landscape.

Our approach is primarily based on the foundational work detailed in - DistilBERT: A distilled version of BERT (Sanh et al., 2019), which serves as the main base paper for this project and demonstrates how a model can retain 97% of BERT's performance while being 40% smaller and 60% faster.

## Problem Statement:

Transformer-based models like BERT have achieved state-of-the-art performance in NLP tasks. However, their high computational complexity restricts deployment in resource-constrained environments. DistilBERT, a compressed version of BERT, provides faster inference with minimal accuracy loss. Nevertheless, its effectiveness in domain-specific applications such as news analysis remains underexplored. Thus, there is a need to evaluate its real-world usability and performance, especially in limited-resource settings.

## Positioning Statement:

"Although DistilBERT offers theoretical efficiency, its real-world applicability in domain-specific NLP tasks remains largely untested. This project addresses that gap by developing a news analysis system capable of summarizing, classifying, and performing sentiment analysis on news articles to mitigate information overload. By integrating DistilBERT and rigorously evaluating its performance under practical constraints, we assess its readiness for deployment in resource-limited environments. Unlike previous studies focused primarily on model design, our approach emphasizes empirical validation and real-world usability."

**Novelty & Significance:**

- **Practical Focus:** This study bridges the gap between theoretical efficiency claims of DistilBERT and its real-world usability in domain-specific tasks.
- **Actionable Insights:** It provides empirical evidence of the trade-offs between speed and accuracy when applying DistilBERT to news-domain applications, offering guidance for its deployment in resource-constrained environments.

## 2.1 Overview of News Article Summarization

The increasing volume of digital media necessitates effective summarization techniques to distill essential information from extensive texts. Summarization methods fall into two primary categories:

**• Extractive Summarization**

Extractive approaches select the most relevant sentences or passages directly from the source text. Early methods relied on frequency-based measures such as Term Frequency-Inverse Document Frequency (TF-IDF) and clustering algorithms. Although computationally efficient, these methods struggled with capturing deeper semantic relationships—a shortcoming that paved the way for neural methods.

**• Abstractive Summarization**

Abstractive summarization generates novel summaries that capture the core meaning of the source content. The breakthrough of Sequence-to-Sequence (Seq2Seq) models with attention mechanisms (Sutskever et al., 2014) enabled the development of more coherent summaries. With the rise of transformer architectures, models such as BERT (Devlin et al., 2018) and, importantly, DistilBERT (Sanh et al., 2019) have advanced the field by providing robust contextual understanding. DistilBERT's efficiency and performance make it a prime candidate for our application, which requires real-time text processing without the heavy computational overhead of larger models.

### 2.1.1 Early Approaches and Statistical Methods

Initial research in summarization employed statistical methods that used heuristics like sentence position, word frequency, and Latent Semantic Analysis (LSA) to rank sentence importance. These methods, although fast, were limited in their ability to grasp the intricate semantics and context necessary for generating truly coherent summaries.

### 2.1.2 Transition to Neural Network Models

The adoption of neural networks marked a significant leap forward. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks began to capture sequential dependencies more effectively. However, these models were challenged by long-range dependencies until attention mechanisms were introduced in Seq2Seq frameworks (Sutskever et al., 2014). This evolution set the stage for transformer architectures, which provide a scalable and powerful solution for understanding and summarizing text.

### 2.1.3 Transformer Models and Their Impact

The introduction of the transformer model in “Attention Is All You Need” (Vaswani et al., 2017) revolutionized NLP by employing self-attention to dynamically weigh contextual relationships. BERT (Devlin et al., 2018) further advanced these ideas by offering deep bidirectional encoding. Our project leverages DistilBERT - as detailed in our base paper by Sanh et al. (2019) - which is distilled to be 40% smaller and 60% faster than BERT, while retaining 97% of its language understanding capabilities. This efficiency is crucial for deploying our application in resource-constrained or on-device scenarios.

## 2.2 Evolution of News Article Classification

Classification of news articles into categories such as politics, sports, technology, and entertainment has similarly advanced from simple rule-based systems to sophisticated deep learning techniques.

### 2.2.1 Rule-Based and Statistical Classification

Early classification efforts relied on rule-based systems that used heuristic rules based on keywords and manually curated features. Subsequently, statistical classifiers such as Naïve Bayes, Support Vector Machines (SVM), and Logistic Regression became popular due to their interpretability, though they were limited in managing high-dimensional data and subtle semantic nuances.

### 2.2.2 Neural Networks and Deep Learning Approaches

The emergence of deep learning revolutionized text classification. Models based on Convolutional Neural Networks (CNNs) and RNNs improved the capture of local and sequential patterns. Yet, it is the introduction of transformer-based models that has dramatically enhanced performance by enabling a deeper understanding of context (Devlin et al., 2018). DistilBERT further optimizes this approach by offering rapid inference and lower computational overhead, which are critical for real-time classification in our application. The transfer learning paradigm (Howard & Ruder, 2018) allows such models to be fine-tuned on relatively small datasets, making them highly effective even with limited labeled examples.

## 2.3 Key Findings in the Literature

A review of the literature reveals several important trends:

**Transformer Dominance:**

Transformer-based models, such as BERT and DistilBERT, have become the standard in both summarization and classification tasks due to their superior contextual understanding (Vaswani et al., 2017; Devlin et al., 2018; Sanh et al., 2019).

**Limitations of Early Methods:**

Although early statistical and rule-based methods were valuable for their simplicity and speed, they lacked the capability to capture the deeper semantic and contextual nuances required for complex NLP tasks.

**Pre-training and Fine-tuning Paradigm:**

The process of pre-training on large text corpora followed by task-specific fine-tuning has proven to be a highly effective approach, as demonstrated by models like DistilBERT (Devlin et al., 2018; Howard & Ruder, 2018).

**Benchmark Evaluation Metrics:**

Summarization is typically evaluated using ROUGE metrics, whereas classification models are assessed using accuracy, precision, and F1-scores. These standardized metrics facilitate an objective comparison of model performance across different studies.

## 2.4 Methodological Gaps and Research Opportunities

Despite recent advancements in natural language processing, key methodological gaps remain particularly in areas that our project aims to address:

**• Resource Efficiency vs. Performance:**

DistilBERT offers a substantial reduction in model size and significantly faster inference times compared to its predecessor, BERT.

However, further validation is needed to assess its real-world usability in resource-constrained environments without compromising accuracy. Specifically, it is essential to evaluate whether the efficiency gains claimed in the DistilBERT research hold true under practical deployment scenarios.

**• Domain and Language Adaptation:**

Although DistilBERT has demonstrated promising results as a general-purpose language model, its performance in domain-specific contexts such as news article analysis remains insufficiently explored. News articles often involve structured reporting, journalistic conventions, and domain-specific terminology that differ from the datasets typically used to evaluate language models. The extent to which DistilBERT can effectively understand and adapt to these domain-specific linguistic features is still an open question, highlighting a critical research opportunity.

## 2.5 Integration of Summarization and Classification

Recent research has increasingly focused on unified frameworks that handle both summarization and classification within a single model architecture.

By sharing representations, these approaches reduce redundancy and improve overall efficiency. Our project leverages DistilBERT-as validated by Sanh et al. (2019) to implement an integrated system capable of performing both tasks effectively.

This approach not only streamlines the processing pipeline but also provides empirical evidence of DistilBERT’s versatility and efficiency in real-world news analysis applications.

## 2.6 Summary and Future Directions

In summary, the literature underscores the transformative impact of transformer-based architectures on NLP. While traditional methods provided important initial insights, modern models especially DistilBERT as presented by Sanh et al. (2019) offer superior contextual understanding, scalability, and computational efficiency. Our project builds upon these advancements to develop an application that performs text summarization and classification in real time. Future research should explore further refinements in domain adaptation, real-time processing, and resource optimization to enhance the performance and deployment of lightweight transformer models in practical, resource-constrained settings.

# Chapter 3: SYSTEM ANALYSIS

In this chapter, we perform an in-depth analysis of the existing systems and methodologies employed for news article summarization and classification, shedding light on their inherent limitations and shortcomings. We then propose a new system design that addresses these challenges, detailing the innovative approaches and algorithms integrated into our project. Additionally, this chapter outlines the essential hardware and software requirements necessary to implement and deploy the new system efficiently.

## 3.1 Analysis of Existing Systems

Over the past decade, numerous systems have been developed to address news article summarization and classification. While early systems relied on statistical and rule-based techniques, subsequent iterations have incorporated machine learning and deep learning approaches. Below, we outline the primary characteristics and limitations of these existing systems.

### 3.1.1 Statistical and Rule-Based Approaches

Early methods for summarization predominantly depended on statistical models that leveraged word frequency, sentence position, and heuristic rules to extract salient content. Methods such as term frequency-inverse document frequency (TF-IDF) and Latent Semantic Analysis (LSA) were widely used. Similarly, rule-based classification systems were designed around pre-defined keywords and simple decision trees, enabling basic filtering of content into categories.

**Limitations:**

- **Loss of Context:** These methods often failed to capture the nuanced meaning of sentences. The use of surface-level features led to summaries that neglected important contextual information.
- **Rigidity:** Rule-based systems are inherently static. They require constant manual updates to accommodate shifting language patterns or emerging topics in news, reducing their adaptability.

- **Insufficient Coherence:** Statistical summarization resulted in output that sometimes appeared disjointed and lacked the narrative flow necessary for understanding complex content.
- **Scalability Concerns:** As the volume of news data increases, these models struggle to maintain performance without frequent recalibration.

### 3.1.2 Machine Learning and Classical Approaches

The introduction of machine learning techniques, particularly traditional classifiers like Naïve Bayes, Support Vector Machines (SVM), and Logistic Regression, marked a significant improvement in classification tasks. These models, often trained on engineered features, provided better accuracy compared to their rule-based counterparts. Meanwhile, extractive summarization approaches began to incorporate supervised learning to rank sentences based on importance.

**Limitations:**

- **Feature Engineering Dependency:** Classical machine learning models required extensive and sometimes error-prone feature engineering. The process was labor-intensive and highly dependent on domain expertise.
- **Limited Semantic Understanding:** These models often struggled with understanding the semantic relationships between words, making it challenging to summarize content accurately when subtle context shifts occurred.
- **Generalization Issues:** They frequently exhibited difficulties in generalizing to previously unseen topics or emerging trends in news, resulting in misclassifications.
- **Static Nature:** Although more adaptive than simple rules, they still exhibited a degree of rigidity and were less effective in adapting to dynamic content without periodic retraining.

### 3.1.3 Neural Network-Based Approaches

The evolution of neural networks, particularly Recurrent Neural Networks (RNNs) and attention-based Sequence-to-Sequence (Seq2Seq) architectures, dramatically improved performance in both summarization and classification. These models could capture context in a sequential manner, leading to more coherent summaries and more accurate text categorization. Early adoption of attention mechanisms allowed systems to dynamically prioritize information in longer texts.

**Limitations:**

- **Computational Overhead:** RNN-based models, despite their improvements, typically require significant computational resources during training and inference, making them impractical for real-time news processing in resource-constrained environments.
- **Vanishing Gradients and Long-Range Dependencies:** Traditional RNNs can struggle with long documents due to issues such as vanishing gradients, limiting their ability to capture dependencies across lengthy texts.
- **Training Time:** These systems necessitate substantial training time, especially when dealing with large datasets, which delays iterative improvement and deployment cycles.
- **Difficulty with Parallelization:** The sequential nature of RNNs limits the potential for parallel processing, thus affecting scalability in high-throughput environments.

### 3.1.4 Transformer-Based Models

The advent of transformer architectures, notably with the introduction of BERT, GPT, and their distilled versions like DistilBERT, has addressed many of the limitations of earlier methods. Transformers leverage self-attention mechanisms to capture long-range relationships and dynamic contextual representations, resulting in superior performance for both summarization and classification tasks.

**Advantages:**

- **Advanced Contextual Awareness:** The self-attention mechanism in transformers allows the model to weigh the relevance of each token in the sequence, thereby generating summaries with greater depth and comprehensibility.
- **Improved Generalization:** Pre-trained transformer models are capable of generalizing well across different genres and topics due to extensive pre-training on diverse corpora.

- **Efficiency through Distillation:** DistilBERT, for example, provides nearly comparable performance to BERT with significantly reduced computational costs. This makes it ideal for real-time applications on moderate hardware configurations.
- **Robust Transfer Learning:** The fine-tuning process permits the model to adapt swiftly to domain-specific news articles using relatively smaller labeled datasets.

  Despite the advancements offered by transformer-based models, challenges remain, particularly regarding the trade-off between maintaining completeness in summarized content and ensuring computational efficiency for real-time applications.

### 3.2 Disadvantages of Existing Systems

While current systems have evolved considerably, a detailed analysis reveals persistent weaknesses that can impact performance in dynamic, real-world applications:

3. **Incomplete Contextual Integration:**

   Many existing extractive models are unable to synthesize content beyond simply selecting key sentences, causing the potential omission of relevant context and leading to summaries that do not capture subtle yet crucial details.

4. **Classification Silos:**

   In systems where summarization and classification are treated as separate tasks, the information flow between these components is not exploited optimally. Consequently, the classification module might not benefit from contextually refined representations generated during summarization, and vice versa.

5. **Insufficient Adaptation to Evolving Data:**

   Traditional methods and even some modern approaches can be limited in their ability to adapt to rapidly changing news environments. The dynamic nature of digital news, with evolving topics and tones, often outpaces the capacity of static models, necessitating frequent retraining and manual updates.

6. **Resource Intensity:**

   Advanced neural network architectures, while powerful, frequently require high-end hardware for training and inference. This makes large-scale deployment and edge computing challenging, particularly in real-time applications where computational resources may be limited to mid-range specifications.

7. **Complexity in Deployment:**

   Systems that integrate multiple algorithms and modules for summarization and classification often encounter difficulties during deployment. Complex interdependencies between system components, diverse library requirements, and maintenance of model pipelines can result in operational hiccups and increased deployment times.

### 3.3 Proposed System: Addressing Limitations and Enhancing Capabilities

Given the aforementioned disadvantages, our proposed system adopts a unified architecture designed to overcome existing limitations in news article summarization and classification. The new system integrates key functionalities with streamlined processes, ensuring that both summarization and classification tasks benefit from shared contextual insights.

#### 3.3.1 Unified Model Architecture

At the heart of our proposed solution is the integration of a pre-trained DistilBERT model, which is fine-tuned for both summarization and classification tasks. This unified model architecture offers several benefits:

- **Shared Contextual Representations:**

  By leveraging the same model architecture for both tasks, the system ensures that the contextual correlations identified during summarization directly inform the classification process. This leads to more precise category assignments and summaries that accurately reflect the core content.

- **Enhanced Efficiency:**

  Using DistilBERT not only reduces computational overhead but also simplifies the system’s operational pipeline. The distilled model maintains performance levels comparable to its larger counterparts while operating within mid-range hardware constraints. This results in faster processing times without compromising on accuracy.

- **Streamlined Data Pipeline:**

  The uniform handling of tokenization, normalization, and feature extraction for both summarization and classification minimize redundancy. A consolidated data pipeline decreases the likelihood of inconsistencies between modules and simplifies debugging and maintenance processes.

### 3.3.2 Advanced Summarization Techniques

To improve upon the shortcomings of purely extractive methods, our system employs a hybrid summarization strategy that incorporates both extractive and abstractive components:

- **Extractive Pre-Selection:**
  Initially, the system reviews the article to identify the most representative sentences based on contextual importance. Techniques such as cosine similarity in the vector space and attention weight aggregation are employed to rank sentences.
- **Abstractive Refinement:**
  Once key sentences have been identified, an additional processing layer rephrases and condenses the content to create a coherent and succinct summary. This step harnesses the generative capabilities of transformer models, ensuring that the output preserves the narrative integrity of the original text.
- **Granularity Adjustment:**
  The summarization module is designed to offer variable levels of detail depending on user preferences. It can generate high-level overviews for quick perusal or more detailed summaries that capture finer nuances of the news report. This flexibility is achieved by adjusting the summarization depth via attention thresholds and configurable summarization lengths.

### 3.3.3 Integrated Classification Module:

Our classification component is equally robust, leveraging fine-tuned transformer models to assign news articles to predefined categories. The integrated approach ensures:

- **Context-Enriched Classification:**
  By using the enhanced contextual data from the summarization process, the classification module can make more informed decisions. This results in higher classification accuracy, as the output categories align closely with the nuanced themes present in the articles.

- **Multi-Label Support:**

  Recognizing that a single article may address multiple subjects, the system is designed to support multi-label classification. This is particularly valuable in the context of news where topics are often interwoven, such as political events with economic implications.
- **Dynamic Learning and Adaptation:**

  The classifier module is equipped with continual learning capabilities to adjust to evolving news topics. By employing techniques such as transfer learning and incremental model updating, the system remains responsive to emerging trends without necessitating complete retraining.

### 3.4 Algorithmic Overview

The core algorithms driving the functionalities of our proposed system are selected to balance performance, accuracy, and computational efficiency. Here, we provide a brief description and scope of these algorithms:

#### 3.4.1 DistilBERT for Contextual Embedding:

- **Description:** DistilBERT is a compact, distilled version of the original BERT model that provides nearly equivalent performance with enhanced computational efficiency. By leveraging self-attention mechanisms, DistilBERT generates robust embeddings that capture semantic relationships within the text.
- **Scope:**
  - **Summarization:** Embeddings generated are used to rank sentences based on contextual similarity and semantic weight.
  - **Classification:** The same embeddings inform the category prediction module, ensuring that subtle contextual cues are not lost during classification.

#### 3.4.2 Hybrid Summarization Algorithm

- **Description:** This approach combines extractive summarization to select key sentences with an abstractive summarization component that refines the text into a coherent summary.

- **Scope:**
  - **Stage One:** Extraction of sentences using attention score aggregation and similarity measures.
  - **Stage Two:** Abstractive refinement where a generative transformer layer rephrases selected content to produce a natural and fluid summary.
  - **Adjustment Mechanism:** Configurable parameters allow the user to choose the level of detail, ranging from ultra-concise to moderately detailed summaries.

### 3.4.3 Multi-Label Classification Strategy

- **Description:** A classification layer built upon transformer-derived embeddings supports multi-label categorization, acknowledging that articles can span various topics simultaneously.
- **Scope:**
  - **Feature Extraction:** Utilizes shared embeddings from DistilBERT.
  - **Prediction Process:** Applies threshold-based multi-label classification that assigns articles to multiple relevant categories using sigmoid activation functions and probability thresholds.
  - **Fine-Tuning:** Ongoing model adjustments via continual learning frameworks ensure adaptability to new and evolving topics in real-time news.

## 3.5 System Requirements

To support the seamless execution of the proposed system, both hardware and software requirements have been carefully considered. The design ensures that implementation is feasible on mid-range specifications, making it accessible to a wide array of institutions and researchers.

### 3.5.1 Hardware Requirements

- **Processor:**
  - Mid-range processors such as Intel Core i5 or AMD Ryzen 5 are sufficient. For higher throughput or real-time processing scenarios, an Intel Core i7 or equivalent is recommended.

- **Memory:**
  - A minimum of 8GB RAM is required, although 16GB is ideal to support parallel processing and large model inference.
- **Graphics Processing Unit (GPU):**
  - While the system is optimized for CPU usage, a mid-range GPU (e.g., NVIDIA GTX 1660 or equivalent) is beneficial during model training and for accelerating inference tasks, particularly with transformer-based models.
- **Storage:**
  - At least 256GB of SSD storage is recommended to manage large datasets, pre-trained model files, and output logs efficiently.
- **Connectivity:**
  - A stable internet connection is necessary for fetching updates from external repositories (e.g., Hugging Face) and for facilitating API calls in the web interface.

### 3.5.2 Software Requirements

- **Operating System:**
  - Compatible with major operating systems including Windows, macOS, and Linux. The system has been primarily developed and tested under Ubuntu and Windows environments.
- **Programming Language:**
  - Python 3.7 or higher is recommended due to its extensive support for libraries in machine learning and NLP.
- **Libraries and Frameworks:**
  - **Transformers and Torch:** Utilized for loading and fine-tuning the DistilBERT model.
  - **Pandas and NumPy:** Critical for data manipulation and processing.
  - **Streamlit:** Provides the framework for developing the user-friendly web interface.
  - **BeautifulSoup4 and Requests:** Used for data extraction and handling web-based content.

  - **Openpyxl:** Employed for managing output in Excel files, ensuring the portability of summarized and classified data.
  - **WordCloud and Matplotlib:** Facilitate visualization tasks such as graphical representations of word frequencies and generated summaries.
  - **NLTK and Summarizer:** Assist in additional text processing and natural language manipulations.
- **Development Environment:**
  - Tools like Jupyter Notebook or Visual Studio Code are recommended for development, along with version control systems (e.g., Git) to manage code revisions and collaborative efforts.
- **Virtual Environment:**
  - It is advisable to implement the system within a virtual environment (e.g., conda or venv) to manage dependencies and package versions effectively.

### 3.6 Advantages of the Proposed System

The new system presents several significant advantages over traditional and earlier systems, as outlined below:

1. **Enhanced-Contextual-Representation:**
   Combining extractive and abstractive techniques allows the model to generate summaries that are
2. **Real-Time Processing Capability:**
   Thanks to the efficiency gains from DistilBERT and streamlined data pipelines, the system is well-suited for real-time applications. This is crucial for scenarios such as breaking news monitoring and emergency management, where speed is of the essence.
3. **Integrated Multi-Label Classification:**
   The system's capability to classify articles into multiple relevant categories simultaneously ensures a comprehensive understanding of the news content, bridging the gap caused by siloed classification approaches in traditional systems.

4. **Adaptability and Scalability:**

   Through continuous learning and transfer learning techniques, the system can adapt dynamically to evolving trends and topics in news media. Its design supports scaling from small datasets to large-scale real-world applications with minimal adjustments.

5. **Resource Efficiency:**

   The use of a distilled transformer model ensures that high performance is achievable even on mid-range hardware specifications. This democratizes access to advanced NLP techniques, enabling educational institutions, startups, and resource-constrained environments to implement cutting-edge news processing systems.

6. **Seamless Deployment:**

   The unified architecture and modular design facilitate easier integration and deployment. With standardized data pipelines and shared model components for both summarization and classification, the system minimizes development overhead and reduces maintenance complexities.

### 3.7 Implementation Considerations:

To realize the envisioned system, several practical implementation strategies have been identified:

- **Pre-Deployment Testing and Tuning:**

  Rigorous stress testing and parameter tuning are essential to balance computational loads, particularly under varying data volumes. Conducting pilot tests using subsets of the THESTAR.COM.MY dataset ensures that the system can handle edge cases and unforeseen anomalies in news data.

- **Iterative Model Improvement:**

  Early deployment should include mechanisms for feedback collection, enabling iterative improvements. Continuous updates through online learning or periodic retraining will ensure the system remains adaptive and resilient in dynamic operational environments.

- **User Interface and Interaction:**

  A well-designed web interface via Streamlit will facilitate accessibility and user-friendliness. Screening for both expert users (e.g., data scientists) and non-expert audiences (e.g., news readers) is critical. This dual-target approach ensures transparency in the summarization process and presents classification results in a clear, digestible format.

- **Security and Data Integrity:**

  Given the web-based interface and data exchange components, integrating robust security protocols is necessary to safeguard user input and maintain data integrity throughout the processing pipeline.

# Chapter 4: SYSTEM DESIGN

This chapter details the structural blueprint of **the "Development and Performance Evaluation of an Application for News Article Summarization, Classification, and Sentiment Analysis using Deep Learning Models"** system. It describes the overall architecture, breaks down system components, and illustrates the data flow and interaction among modules. In doing so, we aim to provide a comprehensive understanding of how the system processes user inputs, performs computations, and generates outputs using advanced NLP techniques with the pre-trained DistilBERT model as its core.

## 4.1 Overall System Architecture

The system is designed with a modular architecture that enhances scalability, maintainability, and flexibility. It comprises several interconnected layers, each playing a distinct role in processing unstructured news data. These layers include the Data Ingestion and Preprocessing Unit, the Core NLP Engine, the Summarization Module, the Classification Module, the Integrated Data Pipeline, and the User Interface (UI) Provider.

This division of labor not only facilitates parallel development and testing but also allows the system to adapt rapidly to evolving requirements in both news summarization and classification tasks.

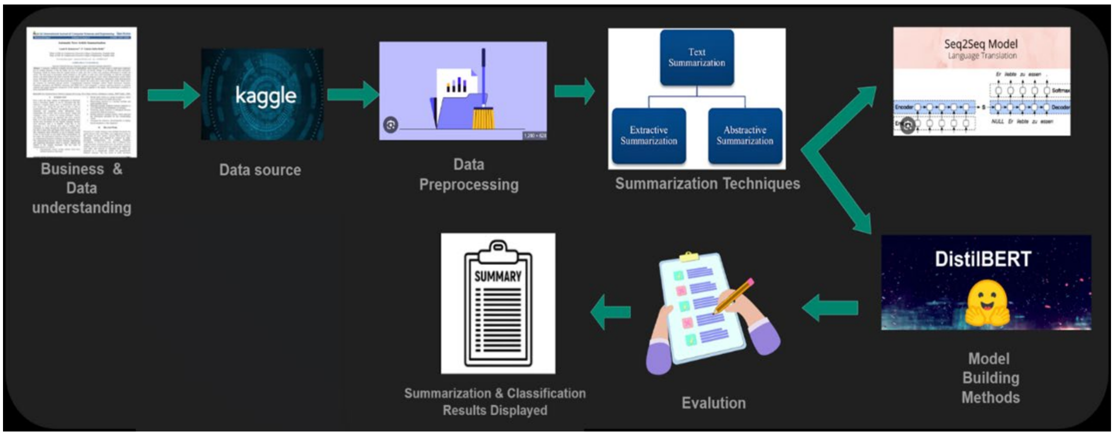


**Fig: Project Architecture**

### 4.1.1 Architectural Layers

**1. Data Ingestion and Preprocessing Unit:**

At the system's entry point, incoming news articles (either from online sources or uploaded documents) are captured via APIs or through file uploads. This module leverages libraries like BeautifulSoup4 and Requests to extract raw text from web-based content. In addition to

extraction, the system standardizes and cleans the data by removing extraneous characters, performing tokenization, normalization, and filtering noise.

**2. Core NLP Engine:**

This layer makes extensive use of the DistilBERT model to generate contextual embeddings that underpin both summarization and classification tasks. By utilizing shared transformer-derived embeddings, the system ensures consistency of contextual representations across both functionalities. The NLP engine is the central processing hub, bridging lower-level text preprocessing with higher-level decision-making tasks.

**3. Summarization Module:**

Designed as a hybrid solution, the summarization module incorporates extractive pre-selection and abstractive refinement. It initially isolates key sentences, ranking them based on attention weights and semantic relevance. Thereafter, a generative transformer component refines these selections into coherent summaries that preserve the article's narrative structure while ensuring brevity and clarity.

**4. Classification Module:**

Built upon the same core transformer embeddings, the classification unit maps news articles to one or more predefined categories. This module is specifically tailored for multi-label classification, ensuring that an article's multifaceted nature is adequately represented. By using probability thresholding (with sigmoid activation in the final layer), the module preserves the nuanced degrees of relevance for each category.

**5. Integrated Data Pipeline:**

The unified data pipeline coordinates the flow of processed data between modules, ensuring that O/P from the summarization module assist the classification process, and vice versa. Through tokenization and feature extraction methods shared across the modules, the pipeline eliminates redundancy and reduces latency.

**6. User Interface Provider:**

A user-friendly web interface developed using Streamlit connects all system components to the end-user. The UI presents summarized texts, classification labels, visualizations such as WordClouds and Matplotlib charts, and even logs any raw outputs that may aid in assessments. Moreover, it supports interactive parameter tuning, which allows users to set the granularity level of summarization, adjust classification thresholds, and monitor real-time performance metrics.

## 4.2 Detailed Component Design

In this section, we delve into the specific design aspects of individual components that collectively enable the seamless operation of the system.

### 4.2.1 Data Ingestion and Preprocessing

The foundation for efficient summarization and classification is robust data ingestion. The preprocessing pipeline comprises several sequential operations conducted on incoming data:

- **Data Extraction:**

  Utilizing modules such as Requests and BeautifulSoup4, the system scrapes online articles and processes locally stored files. These tools ensure that web-based news articles are captured accurately and stored in text format.
- **Text Normalization:**

  In this step, text data is subject to cleaning operations such as lowercasing, punctuation removal, and stop-word filtering primarily using NLTK. This normalization is crucial in minimizing noise that could potentially skew the performance of the downstream transformer models.
- **Tokenization:**

  Tokenization is performed with a tokenizer that is optimized for the DistilBERT architecture. The tokenizer converts raw text into appropriate sequences of tokens, preparing them for efficient ingestion by the transformer model.

- **Feature Extraction:**
  Supplementary features, including metadata like publication time, source, and article length, are computed. These features assist in fine-tuning both summarization and classification tasks, as well as in performing exploratory data analysis.

### 4.2.2 Core NLP Engine and Contextual Embeddings

At the heart of this system lies the DistilBERT model, a distilled version of BERT renowned for its computational efficiency and high performance. The core NLP engine is responsible for processing text data and generating contextual embeddings. These embeddings are essential because they capture the semantic relationships and contextual cues present within the articles.

- **Embedding Generation:**
  The DistilBERT model processes the tokenized text to generate dense vector representations. Such representations facilitate both the selection of summary sentences and the accurate determination of article categories.
- **Model Fine-Tuning:**
  The pre-trained DistilBERT is fine-tuned on the THESTAR.COM.MY dataset, enabling the model to adapt to the specific vocabulary, stylistic nuances, and themes prevalent in news articles. Fine-tuning uses transfer learning techniques to boost efficiency without extensive retraining from scratch.
- **Shared Contextual Representations:**
  Owing to the integrated architecture, the same embeddings are used by both the summarization and classification modules. This shared representation plays a vital role in ensuring consistency and enhancing overall system performance by allowing the classifier to derive context insights from summarization-derived signals.

### 4.2.3 Summarization Module

The summarization module employs a dual-stage mechanism:

- **Extractive Pre-Selection:**
  The module begins by scoring each sentence in a news article based on its contextual importance. Techniques such as computing cosine similarity with a

global sentence vector and aggregating attention weights are employed to extract a subset of the most relevant sentences.

- **Abstractive Refinement:**
  Once key sentences are identified, an auxiliary transformer-based decoder rephrases and consolidates these sentences to generate a coherent and fluid narrative summary. This abstractive component ensures that while key points are preserved, the final summary is not a mere patchwork of disjointed sentences. Instead, it forms a concise narrative that respects the article's original context.
- **Configurable Granularity:**
  Users can adjust the granularity of the summarization process via the Streamlit interface. For example, using slider controls or text input, users can select between ultra-concise summaries suitable for quick insights or extended summaries that offer a more detailed perspective.

### 4.2.4 Classification Module

The classification module is engineered to support multi-label categorization, acknowledging that news articles frequently straddle several topics:

- **Label Prediction:**
  Using the embeddings derived from the NLP engine, the module applies a series of dense layers culminating in a sigmoid activation to yield probability scores for each potential label. These probabilities represent the likelihood that a given article belongs to specific categories such as politics, sports, entertainment, technology, and more.
- **Thresholding Mechanism:**
  A configurable threshold setting ensures that only categories exceeding a predetermined probability value are selected. This prevents classifications based on low-confidence predictions. The Streamlit interface further allows users to adjust these thresholds to fine-tune performance based on different use cases or evolving dataset characteristics.
- **Integration with Summarization:**
  Importantly, the classification module leverages reformulated representations from the summarization process. This synergy offers a refined decision-making

basis, where nuanced textual features extracted during summarization are also utilized for category predictions. This leads to improved accuracy and consistency, especially in articles with multiple themes.

### 4.2.5 User Interface and Visualization

The user-facing component is built on Streamlit, which is favored for its simplicity in creating highly interactive web applications. The UI comprises several functional sections:

- **Input Section:**
  Users can either submit URLs, upload files, or paste raw text into a dedicated input area. This section also includes options for choosing summarization detail, adjusting classification thresholds, and triggering processing pipelines.

- **Output Display:**
  Processed outputs are displayed in a clear and intuitive manner. Summaries are highlighted along with their corresponding classification labels. Visual aids such as WordClouds, bar charts, and line graphs (created using Matplotlib) offer insights into word distributions, category frequency, and processing time, empowering users with both qualitative and quantitative feedback.

- **Interactive Controls:**
  The interface includes interactive sliders, dropdown menus, and toggle buttons that allow end-users to adjust parameters dynamically. This facilitates experiments with different levels of summarization and classification granularity, as well as immediate feedback on how these adjustments impact system output.

### 4.3 Data Flow Diagram and System Interaction

To illustrate the interconnection among modules and the flow of data within the system, we provide a detailed description of a Data Flow Diagram (DFD). While a graphical DFD is ideal, the following narrative provides clarity.

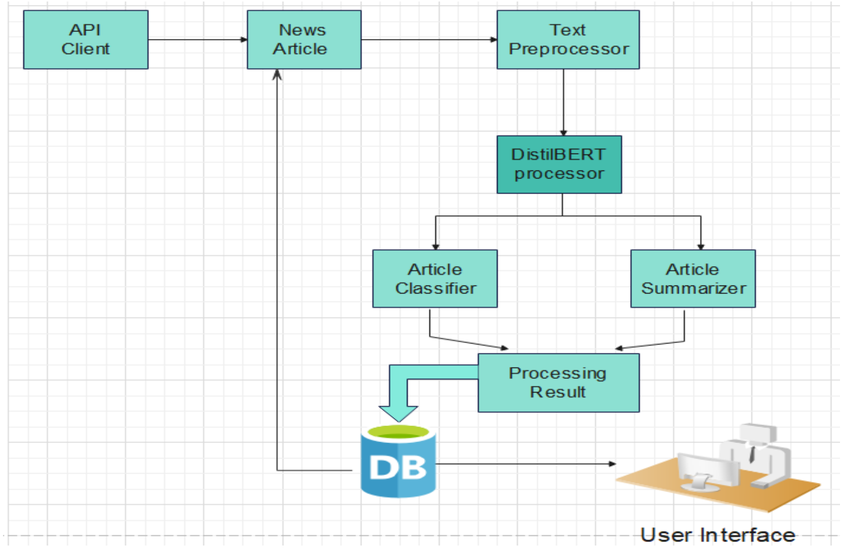


**Fig: Data Flow Diagram**

### 4.3.1 Data Flow Description

7. **Input Data Reception:**

   The system begins by receiving raw news articles via the UI. These articles are forwarded to the Data Ingestion Unit, where they are parsed and preprocessed.

8. **Preprocessing Pipeline:**

   Preprocessing activities include text cleaning, tokenization, and metadata extraction. The resulting clean text is sent to the Core NLP Engine.

9. **Contextual Embedding Generation:**

   The Core NLP Engine utilizes DistilBERT to generate embeddings. A portion of these embeddings is directed to the Summarization Module, while the full set is concurrently relayed to the Classification Module.

10. **Summarization Processing:**

    The Summarization Module operates in two phases (extractive and abstractive). During the extractive phase, an attention-based ranking filter selects key sentences, and in the abstractive phase, a transformer decoder produces a refined summary. The summary is stored and also passed on to the Classification Module to support context enrichment.

11. **Classification Processing:**

    With the help of shared embeddings and refined contextual input from summarization, the Classification Module computes probabilities for each predefined category. It then applies thresholding to finalize the multi-label output.

12. **Consolidated Output Generation:**

    Both the summary and classification labels are packaged into a unified output structure. This consolidated output is transported back to the UI layer, where it is rendered for user review.

13. **Feedback Loop:**

    Users are given the option to adjust parameters through the web interface. This feedback (e.g., changing summarization granularity) re-triggers parts of the pipeline, thereby enhancing the system's iterative learning and fine-tuning.

### 4.5 Summary of System Design Elements

The architectural design of the "Development and Performance Evaluation of an Application for News Article Summarization, Classification, and Sentiment Analysis using Deep Learning Models" system is built on robust, modular principles:

- It leverages a combination of pre-trained transformer models (DistilBERT) for generating rich contextual embeddings.
- A dual-stage hybrid summarization process—comprising extractive and abstractive methods ensures highly coherent summaries.
- The integrated multi-label classification module uses shared representations from the summarization stage, creating a synergy that improves the accuracy and overall system efficacy.
- A streamlined data flow and interactive UI powered by Streamlit allow seamless exchanges between users and the underlying algorithms.
- Finally, the system's design takes into account not only the algorithmic capabilities but also practical requirements such as hardware specifications, deployment considerations, and security measures.

  This comprehensive design ensures that users from students and educators to professionals in computer science and data analytics experience a system that is both technically robust and highly adaptable to varying operational scenarios.

# Chapter 5: UML Diagrams

In this chapter, we delve into the use of UML (Unified Modeling Language) diagrams to visually represent the various components and interactions within the "Development and Performance Evaluation of an Application for News Article Summarization, Classification, and Sentiment Analysis using Deep Learning Models" system. UML diagrams provide a standardized way to depict system structure, behavior, and interactions.

## 5.1 Class Diagram

This section outlines the key classes and their interactions in the News Article Summarization & Classification system.

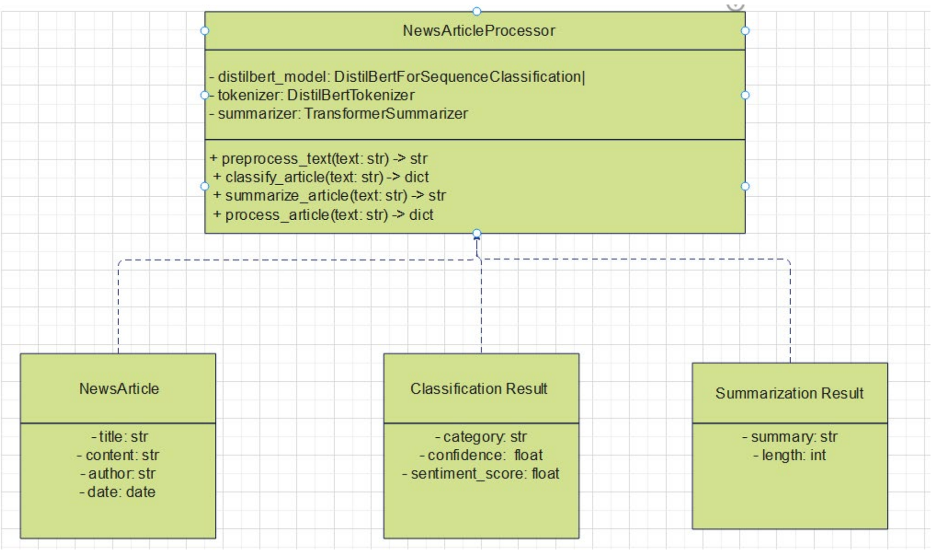

NewsArticleProcessor
- distilbert_model: DistilBertForSequenceClassification|
- tokenizer: DistilBertTokenizer
- summarizer: TransformerSummarizer
+ preprocess_text(text: str) -> str
+ classify_article(text: str) -> dict
+ summarize_article(text: str) -> str
+ process_article(text: str) -> dict
NewsArticle
- title: str
- content: str
- author: str
- date: date
Classification Result
- category: str
- confidence: float
- sentiment_score: float
Summarization Result
- summary: str
- length: int

Fig: Class Diagram

| Class | Attributes | Methods | Role |
|---|---|---|---|
| NewsArticle | articleID, title, content | cleanText(), tokenize() | Stores article data |
| Preprocessor | stopWords, punctuationList | normalizeText(), removeNoise() | Cleans input |
| DistilBERTEngine | preTrainedModel | generateEmbeddings() | Converts text into embeddings |
| SummarizationModule | granularityLevel | extractKeySentences() | Summarizes content |
| ClassificationModule | classificationModel | predictLabels() | Assigns topic labels |

| | | | |
|---|---|---|---|
| UserInterface | uiElements | displayResults() | Shows output |

### 5.2 Sequence Diagram

The sequence diagram illustrates the order of interactions between system components during news article processing.

- User submits an article → UI sends data for processing.

- Preprocessing → Text is cleaned and tokenized.

- Embedding Generation → DistilBERT extracts features.

- Summarization → Extracts key sentences.

- Classification → Assigns category labels.

- Display Results → UI presents the summary and labels.

### 5.3 Use Case Diagram

This section describes key user interactions with the system.

| Use Case | Actor | Description |
|---|---|---|
| Submit Article | User | Inputs an article |
| Preprocess Data | System | Cleans and tokenizes |
| Generate Summary | System | Extracts key content |
| Classify Article | System | Assigns categories |
| Display Results | System | Shows summary & labels |

### 5.4 Flowchart

The flowchart outlines the high-level processing flow of the system.

- Start → User submits an article.

- Preprocessing → Clean & tokenize text.

- Embedding Generation → Convert text into vectors.

- Summarization & Classification (Parallel processing).

- Merge & Display Results.

- End / Adjust Parameters (Optional).

# Chapter 6: MODULES

This chapter outlines the core modules of the **Development and Performance Evaluation of an Application for News Article Summarization, Classification, and Sentiment Analysis using Deep Learning Models**. The project is designed using a modular architecture to ensure efficiency, scalability, and ease of development. The Streamlit-based WEB UI serves as the user interface, while the **DistilBERT-powered NLP engine** handles summarization and classification.

## 6.1 Overview of System Modules

The system consists of the following key modules:

**Core NLP Engine**: Uses DistilBERT to process and generate contextual embeddings for both summarization and classification.

- **Summarization Module**: Condenses text using extractive and abstractive techniques.
- **Classification Module**: Categorizes news articles into relevant topics using multi-label classification.
- **Sentiment Analysis Module**: Analyses the sentiment of news articles using DistilBERT.

## 6.2 WEB UI Provider

### 6.2.1 Role of the WEB UI

The Streamlit-based WEB UI serves as the user-friendly interface, allowing users to interact with the system without needing technical expertise. It provides:

Accessibility: Users can interact with the system via a web browser without additional software.

Interactivity: Users can adjust settings (summary length, classification thresholds) dynamically.

Visualization: Generates summaries, classification labels, and graphical insights.

### 6.2.2 UI Components

1. Input Panel
    - Paste text or upload a file (news articles in text format).

   - URL entry (web scraping with BeautifulSoup4).
2. Processing Controls
   - Summary granularity slider (adjust summary length).
   - Classification threshold settings.
3. Output Display
   - Summarization Results: The generated summary is displayed.
   - Classification Labels: Categories with confidence scores.
   - Visual Analytics: WordCloud and Matplotlib charts.
4. Additional Features
   - Reprocess Button: Users can modify settings and re-run the analysis.
   - Download Option: Export results in text or Excel format.

### 6.3 Integration with Other Modules

The WEB UI interacts seamlessly with:

- Data Preprocessing: Cleans and prepares text before NLP processing.
- Core NLP Engine: Uses DistilBERT embeddings for summarization & classification.
- Summarization & Classification Modules: Generates concise summaries and topic labels.
- Visualization: Presents keyword frequency and category distribution using graphs.

### 6.4 Technical Considerations

#### 6.4.1 Performance Optimization

- Uses efficient tokenization to handle large news articles.
- Minimizes latency through optimized embedding generation.

#### 6.4.2 Scalability & Deployment

- Deployable via Docker for scalable cloud-based processing.
- Supports multi-user access for broader usability.

#### 6.4.3 Security Measures

- Encrypted API endpoints for data exchange.
- User authentication for restricted access.

### 6.5 Advantages of Modular Architecture

- User-Friendly: Simple UI with easy controls.
- Transparency: Users can see how their input is processed.
- Customizability: Adjustable summary & classification parameters.
- Cross-Platform Access: Works in any browser.

# Chapter 7: TECHNOLOGY DESCRIPTION

This chapter provides an overview of the key technologies used in the **Development and Performance Evaluation of an Application for News Article Summarization, Classification, and Sentiment Analysis using Deep Learning Models** project. The system is primarily built on Python, a widely used programming language in Natural Language Processing (NLP) due to its simplicity and rich library ecosystem.

## 7.1 Python and Its Role

### 7.1.1 Overview of Python

Python was created by Guido van Rossum in 1991 and has grown into one of the most popular programming languages due to its simplicity, readability, and versatility. It is widely used in areas like AI, machine learning, and NLP because of its extensive support for data processing and model training.

### 7.1.2 Why Python is Used?

- Easy to learn and use – Simple syntax improves readability.
- Large ecosystem – Rich set of libraries supports NLP, machine learning, and data analysis.
- Cross-platform compatibility – Works on Windows, macOS, and Linux.
- Community support – Well-documented and widely used in research and industry.

### 7.1.3 Setting Up Python for the Project

To work on this project, the following steps are required:

1. Download Python (Version 3.7 or higher) from the official website.
2. Install essential libraries like Pandas, NumPy, Transformers, and NLTK.
3. Set up a virtual environment to manage dependencies.

## 7.2 How Python Powers the Project

Python plays a crucial role in data collection, text processing, model execution, and visualization.

### 7.2.1 Data Collection & Preprocessing

- Web Scraping: Extracts news articles using Requests and BeautifulSoup4.
- Text Cleaning: Libraries like NLTK remove unnecessary characters, stop words, and tokenize text for processing.
- Data Management: Pandas organizes and processes large datasets efficiently.

### 7.2.2 Summarization & Classification

- Summarization: Uses both extractive and abstractive techniques to condense news content.
- Classification: DistilBERT, a deep learning model, categorizes news into relevant topics such as Politics, Sports, and Technology.

### 7.2.3 Visualization & Output Generation

- Interactive Web UI: Streamlit provides an easy-to-use web interface for inputting articles and receiving results.
- Graphical Representations: WordCloud and Matplotlib generate charts, graphs, and keyword clouds to visually represent news trends.
- Data Export: Openpyxl helps generate Excel reports for further analysis.

## 7.3 Key Technologies & Libraries

Python’s capabilities are extended through various specialized libraries:

- Pandas & NumPy – Handle and process large datasets.
- Requests & BeautifulSoup4 – Fetch and extract news articles from web pages.
- Transformers & PyTorch – Enable deep learning-based text processing.
- Matplotlib & WordCloud – Provide visual insights through charts and word clouds.
- NLTK & Summarizer – Enhance text processing and summarization.

## 7.4 System Performance & Deployment

- Modular Architecture – Components are independently designed for easy upgrades and modifications.
- Optimized Performance – Uses lightweight models like DistilBERT for fast and efficient processing.
- Scalability – Can be deployed on cloud services or local machines.

# CHAPTER 8: SYSTEM TESTING

System testing ensures that our **News Article Summarization and Classification Using NLP** project functions as expected. The focus is on verifying individual components, interactions between modules, and overall system performance. The key testing strategies include **Unit Testing, Integration Testing, Functional Testing**, and **Performance Evaluation**.

## 8.1 Overview of Testing Methodologies

Testing is carried out at different levels:

- **Unit Testing:** Ensures individual functions, such as text preprocessing, tokenization, embedding generation, summarization, and classification, work correctly.
- **Integration Testing:** Validates that data flows seamlessly across modules— from **user input (Streamlit UI) → DistilBERT NLP engine → summarization & classification models → output display**.
- **Functional Testing:** Confirms that the system generates meaningful summaries and correct classifications per expected use cases.
- **Performance Testing:** Assesses response times, system efficiency, and robustness under different loads.

  Both **White Box** (code-level validation) and **Black Box** (output-based validation) testing approaches are applied.

## 8.2 Unit Testing

### 8.2.1 Objectives

Unit testing targets critical functions such as:

- **Preprocessing**: Cleaning and tokenizing text.
- **DistilBERT Embedding Generation**: Ensuring embeddings are properly generated.
- **Summarization Module**: Extracting key sentences and producing meaningful abstractive summaries.
- **Classification Module**: Ensuring multi-label classification produces accurate labels.

### 8.2.2 Implementation

Unit tests are implemented using **Python's unittest and PyTest** frameworks. Sample test cases:

- Ensuring tokenization removes punctuation and normalizes text.
- Checking that the DistilBERT model generates correct embedding dimensions.
- Comparing generated summaries with reference summaries.
- Validating classification probabilities for expected categories.

## 8.3 Integration Testing

### 8.3.1 Objectives

Integration testing ensures different modules work cohesively:

- **Preprocessing → DistilBERT NLP Engine**: Confirming tokenized inputs match model expectations.
- **Summarization Pipeline**: Ensuring extractive and abstractive summarization produces relevant content.
- **Classification Module**: Validating the embeddings correctly map to predefined categories.
- **Streamlit UI → Backend Cohesion**: Checking that user inputs are correctly processed and displayed.

### 8.3.2 Implementation

Integration testing simulates **real-world scenarios**, such as:

- Uploading news articles via the **Streamlit UI** and verifying correct processing.
- Checking that classification labels **match expected outputs**.
- Testing the summarization pipeline with **varying article lengths**.

**Automation Tools:**

- **Selenium** for UI testing.
- **GitHub Actions** for continuous integration testing.

## 8.4 Functional Testing

### 8.4.1 Objectives

Functional testing ensures the **end-to-end** system meets expectations:

- **Summarization Accuracy**: The system generates **coherent, concise summaries**.

- **Classification Reliability**: Articles are categorized **correctly based on content**.
- **Streamlit UI Responsiveness**: User interactions (e.g., adjusting summarization length) produce expected results.

### 8.4.2 Methodology

Functional testing scenarios include:

- **User Acceptance Testing (UAT):** Evaluating system performance with real users.
- **End-to-End Testing:** Submitting sample articles and verifying the entire pipeline functions correctly.
- **Performance Benchmarking:** Ensuring the **ROUGE score** and **classification accuracy** meet predefined metrics.

## 8.5 White Box & Black Box Testing

### White Box Testing

- Code validation for **preprocessing, embedding generation, and summarization logic**.
- Profiling execution time for performance optimization.

### Black Box Testing

- Testing without knowing internal workings—ensuring **correct outputs** for various input articles.
- Security testing to **prevent vulnerabilities** in user input handling.

## 8.6 Performance & Regression Testing

### 8.6.1 Robustness Testing

- **Invalid Input Handling:** Ensuring system does not crash with incomplete or corrupted news articles.
- **Stress Testing:** Evaluating model response times under high input loads.

### 8.6.2 Regression Testing

- **Automated test suites** validate existing functionalities when new features are added.
- **Continuous Integration (CI/CD)** ensures stable updates.

### 8.7 Tools & Frameworks

- **Unit & Integration Testing:** PyTest, unittest
- **UI Testing:** Selenium, Streamlit testing utilities
- **Continuous Testing:** GitHub Actions
- **Performance Profiling:** Python profiling tools, benchmarking frameworks

# Chapter 9: OUTPUT SCREENSHOTS

In this chapter, we provide a detailed overview of the various output screenshots that demonstrate the functionality of the "Development and Performance Evaluation of an Application for News Article Summarization, Classification, and Sentiment Analysis using Deep Learning Models" system. These screenshots validate the performance of both back-end processing and user interaction components. They showcase how the system transforms raw news articles into concise summaries and accurate categorization labels that are presented through an intuitive web-based interface. The following sections describe each type of output screenshot, explaining its significance and highlighting the key features illustrated.

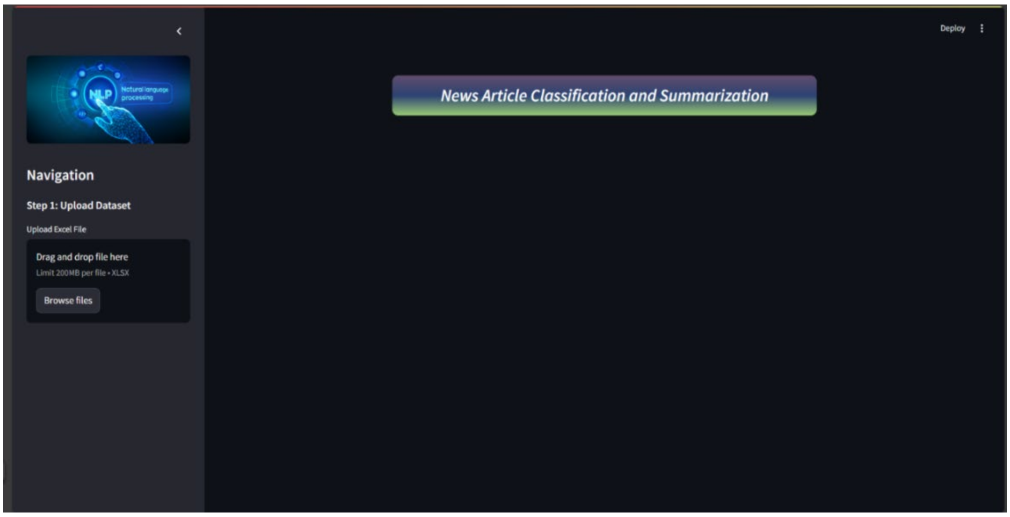
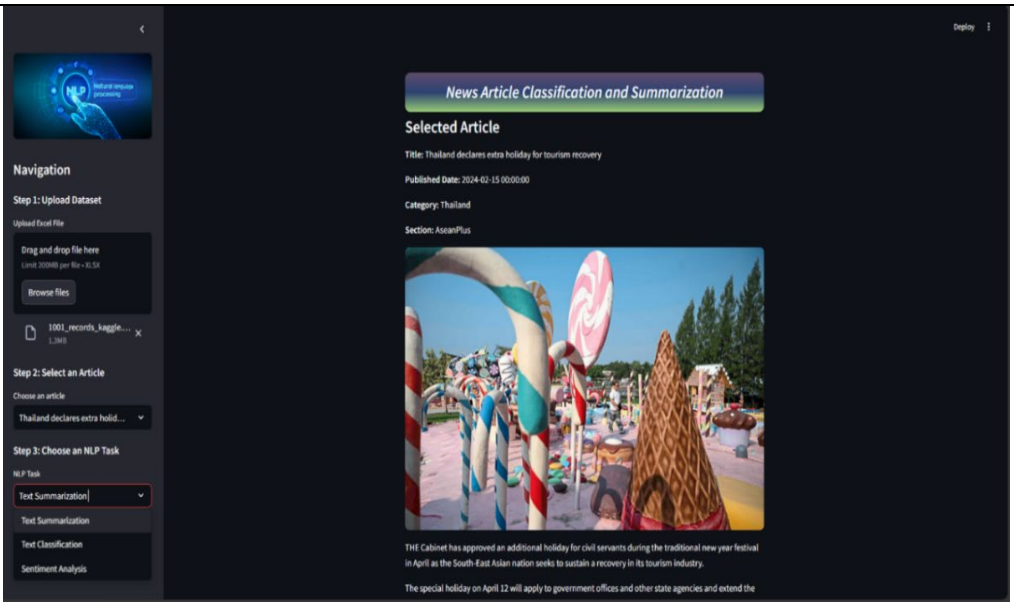

Deploy
NLP
Natural language processing
Navigation
Step 1: Upload Dataset
Upload Excel File
Drag and drop file here
Limit 200MB per file • XLSX
Browse files
News Article Classification and Summarization

**Fig 1: Web UI Landing Page**



**Title:** Thailand declares extra holiday for tourism recovery

**Published Date:** 2024-02-15 00:00:00

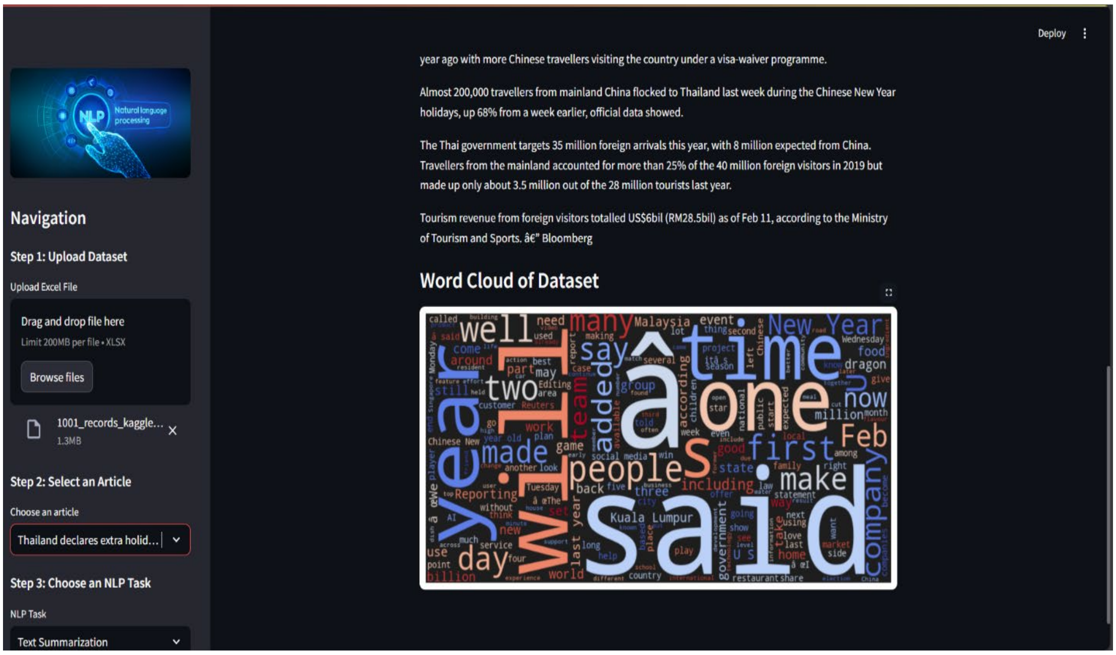
**Category:** Thailand

**Section:** AseanPlus

THE Cabinet has approved an additional holiday for civil servants during the traditional new year festival in April as the South-East Asian nation seeks to sustain a recovery in its tourism industry.

The special holiday on April 12 will apply to government offices and other state agencies and extend the

**Fig 2: A News Article is Loaded from The Dataset**



Navigation

Step 1: Upload Dataset

Upload Excel File

1001_records_kaggle...
1.3MB

Step 2: Select an Article

Choose an article: Thailand declares extra holid...

Step 3: Choose an NLP Task

NLP Task: Text Summarization

year ago with more Chinese travellers visiting the country under a visa-waiver programme.

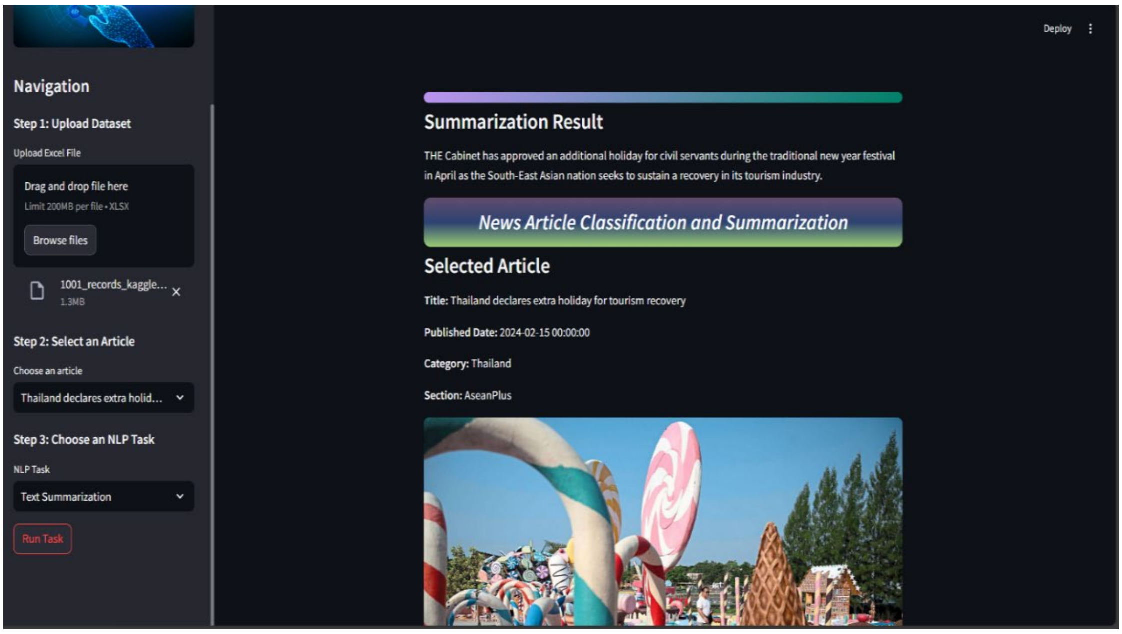
Almost 200,000 travellers from mainland China flocked to Thailand last week during the Chinese New Year holidays, up 68% from a week earlier, official data showed.

The Thai government targets 35 million foreign arrivals this year, with 8 million expected from China. Travellers from the mainland accounted for more than 25% of the 40 million foreign visitors in 2019 but made up only about 3.5 million out of the 28 million tourists last year.

Tourism revenue from foreign visitors totalled US$6bil (RM28.5bil) as of Feb 11, according to the Ministry of Tourism and Sports. â€" Bloomberg

Word Cloud of Dataset

**Fig 3: Displaying the Word Cloud**



Navigation

Step 1: Upload Dataset

Upload Excel File

1001_records_kaggle...
1.3MB

Step 2: Select an Article

Choose an article: Thailand declares extra holid...

Step 3: Choose an NLP Task

NLP Task: Text Summarization

Summarization Result

THE Cabinet has approved an additional holiday for civil servants during the traditional new year festival in April as the South-East Asian nation seeks to sustain a recovery in its tourism industry.

*News Article Classification and Summarization*

Selected Article

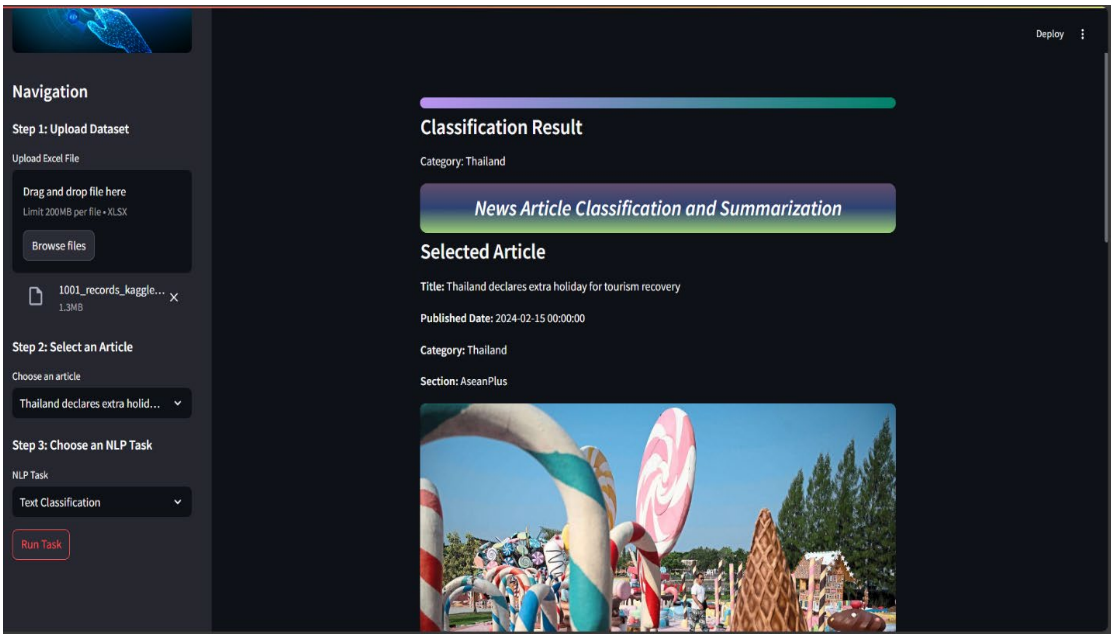
**Title:** Thailand declares extra holiday for tourism recovery

**Published Date:** 2024-02-15 00:00:00

**Category:** Thailand

**Section:** AseanPlus

**Fig 4: Displaying Text Summarization Results**



Classification Result

Category: Thailand

News Article Classification and Summarization

Selected Article

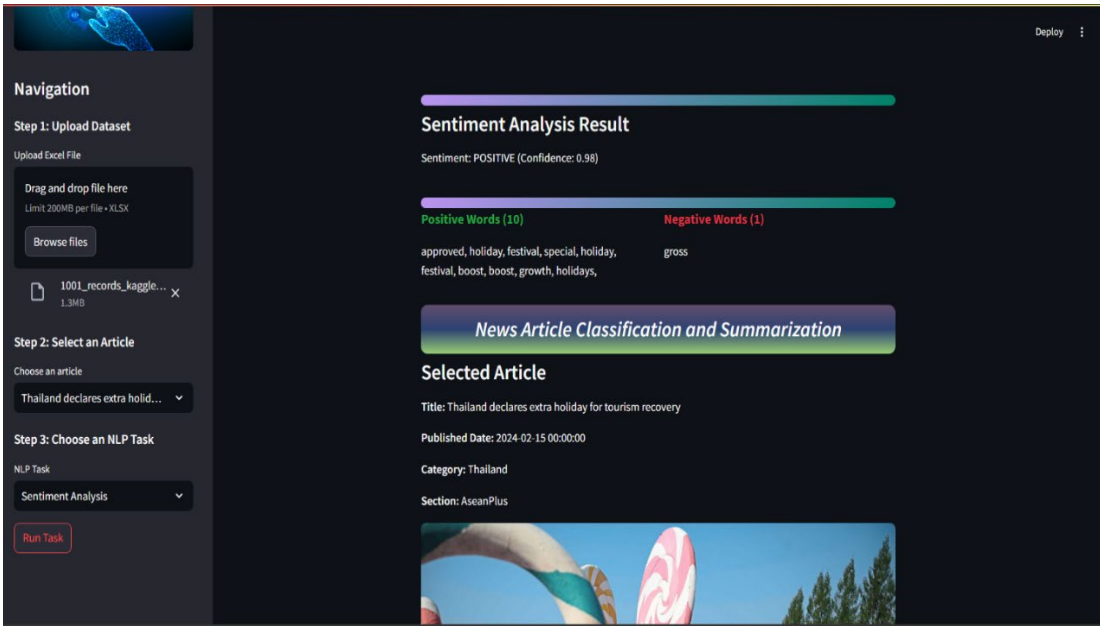
**Title:** Thailand declares extra holiday for tourism recovery

**Published Date:** 2024-02-15 00:00:00

**Category:** Thailand

**Section:** AseanPlus

**Fig 5: Displaying Text Classification Results**



Sentiment Analysis Result

Sentiment: POSITIVE (Confidence: 0.98)

| Positive Words (10) | Negative Words (1) |
|---|---|
| approved, holiday, festival, special, holiday, festival, boost, boost, growth, holidays, | gross |

News Article Classification and Summarization

Selected Article

**Title:** Thailand declares extra holiday for tourism recovery

**Published Date:** 2024-02-15 00:00:00

**Category:** Thailand

**Section:** AseanPlus

**Fig 6: Displaying Sentimental Analysis Results**

## 9.1 Summary Display Interface

A central screenshot shows the output page after a news article has been processed:

- **Formatted Text Summaries:** The summary section features neatly formatted text containing upper, lower, and bullet-point elements for enhanced readability. Important phrases are highlighted, and the summary is displayed in a scrollable text box that allows users to view the entire content with ease.

- **Dynamic Information Panels:** Beside the summary text, captions display information such as the article's title, publication date, and the number of key sentences extracted. Users can compare this output with their understanding of the original text to gauge consistency.

### 9.2 Classification and Labeling Outputs

Another prominent screenshot focuses on the classification results:

- **Multi-Label Presentation:** The screen is divided where classification labels are displayed with accompanying confidence percentages. Each label (for example, "Politics," "Technology," or "Economy") is color-coded. This visual distinction adds clarity and helps users quickly identify the dominant topics discussed in the article.
- **Interactive Label Adjustments:** The interface not only shows static labels but also features buttons or drop-down menus that allow users to adjust thresholds or re-run classification with modified settings. This interactivity fosters a sense of control and flexibility among remote users.
- **Visual Analytics:** Graphs generated from Matplotlib and WordCloud images complement the textual output, offering insights into the frequency of key terms and the overall thematic distribution within the article. These visuals aid in confirming that the classification aligns with the actual subject content.

### 9.3 Detailed Visualizations and Data Representations

Screenshots also present additional visual assets aimed at enhancing user comprehension through interactive analytics:

- **WordCloud Generation:** A snapshot displays an autogenerated WordCloud representing the most frequent terms found within the summarized text. This visualization is particularly useful for remote users to quickly grasp the context and primary emphasis of the article.

## 9.4 Evaluation Output:

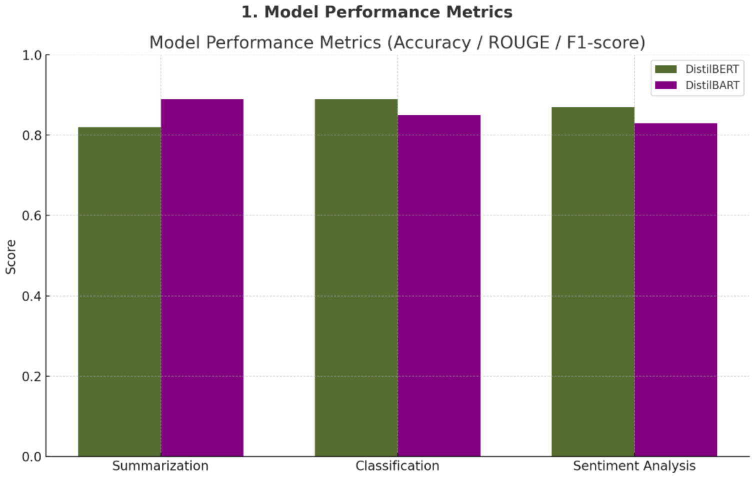


Fig 7: Model Performance Metrics

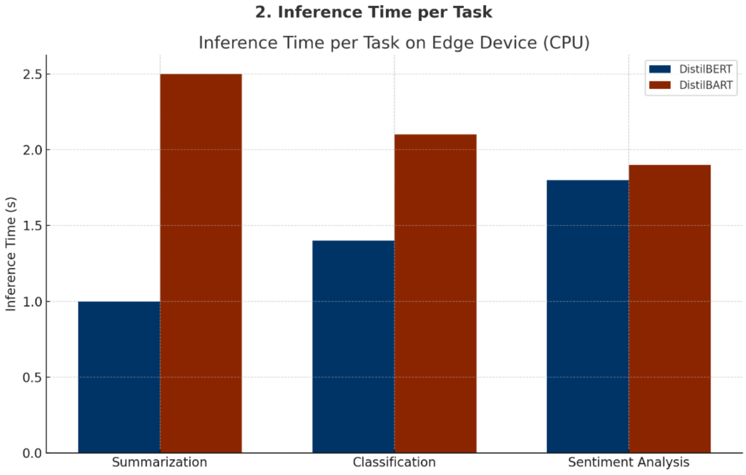


Fig 8: Inference Time per Task

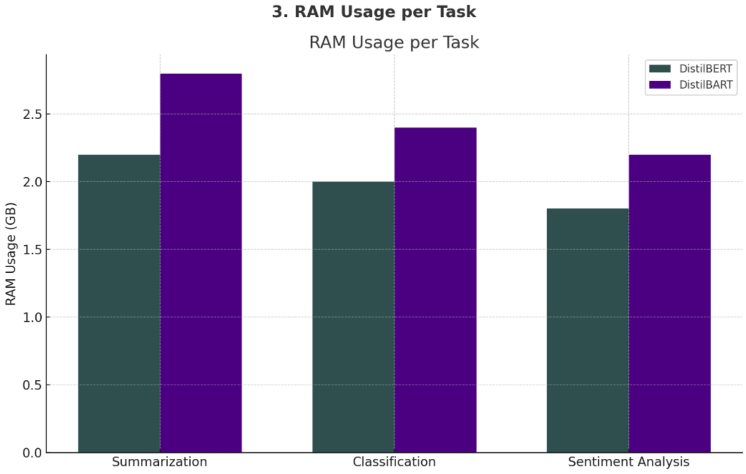


Fig 9: Ram Usage per Task

**Descriptions:**

**Fig 7:** The first bar chart compares the performance of DistilBERT and DistilBART across three tasks: Summarization, Classification, and Sentiment Analysis. DistilBART shows a slight edge in Summarization, while DistilBERT performs better in Classification and Sentiment Analysis. This indicates that DistilBERT may be more versatile across tasks, while DistilBART is more tailored for summarization.

**Fig 8:** The second chart presents the inference time for each model on an edge device using CPU. DistilBERT consistently shows faster inference times across all tasks compared to DistilBART, highlighting its efficiency and suitability for deployment in resource-constrained environments.

**Fig 9:** The third chart depicts the RAM usage of each model per task. DistilBERT uses less RAM in all scenarios, making it a more lightweight and resource-efficient model for edge devices. DistilBART, while more accurate for summarization, demands higher memory usage.

# Chapter 10: CONCLUSION

The "**Development and Performance Evaluation of an Application for News Article Summarization, Classification, and Sentiment Analysis using Deep Learning Models**" project developed a deep learning-based application that automates the summarization, classification, and sentiment analysis of news articles using DistilBERT.

The system efficiently processes lengthy news content into concise summaries, accurately categorizes articles into relevant topics, and analyzes their sentiment to provide deeper insights. By leveraging the speed and performance of DistilBERT, the application proves suitable for real-time use in environments with limited computational resources.

Through this work, we demonstrate the practical applicability of lightweight transformer models in handling domain-specific NLP tasks and offer a step toward more accessible and efficient news consumption. Future improvements may include support for multiple languages and real-time news integration.

---

**Final Thoughts**

As NLP technologies continue to evolve, we anticipate that the insights gained from this project will inform future innovations in automated news processing and other domain-specific applications. Our ongoing commitment to empirical validation and iterative improvements promises to enhance the system's capabilities further, empowering users with reliable and scalable tools in the digital news landscape.

Thank you for engaging with our work. We look forward to its continued impact and further advancements in the field of natural language processing.

---

# Chapter 11: FUTURE ENHANCEMENTS

As **NLP Technologies** continue to evolve, there are several ways to improve the **"Development and Performance Evaluation of an Application for News Article Summarization, Classification, and Sentiment Analysis using Deep Learning Models"** system. This chapter briefly highlights key areas for future enhancements, focusing on **model improvements, user experience, scalability, and ethical considerations**.

### 11.1 Model Enhancements

- **Further Fine-Tuning**: Fine-tune **DistilBERT** on domain-specific datasets (e.g., political, financial, or scientific news) to **enhance classification accuracy**.
- **Hybrid Summarization Approach**: Combine **extractive and abstractive techniques** (e.g., integrating DistilBERT with sequence-to-sequence models for better summary coherence).

### 11.2 Scalability & Deployment Enhancements

- **Cloud & Containerization**: Deploying the system using **Docker & cloud platforms (AWS, Google Cloud)** for better scalability.
- **Monitoring Dashboards**: Implementing **real-time performance monitoring** to track system efficiency and user engagement.

# REFERENCES:

[1] **Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019).** DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

[2] **Chen, C., & Zhu, J. (2018).** Deep Learning-Based Text Summarization: The State of the Art and Future Directions. *arXiv preprint arXiv:1811.01488.*

[3] **Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018).** BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long and Short Papers)* (pp. 4171–4186).

[4] **Dey, S., Ghosh, S., & Singh, A. P. (2019).** A Comprehensive Survey of Text Summarization Algorithms. *AI Review, 52*(1), 793–829. doi:10.1007/s10462-018-9704-7.

[5] **Howard, J., & Ruder, S. (2018).** Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 328–339).

[6] **Sutskever, I., Vinyals, O., & Le, Q. V. (2014).** Sequence to Sequence Learning with Neural Networks. *Advances in Neural Information Processing Systems, 27*, 3104–3112. Retrieved from arXiv:1409.3215.

[7] **Vaswani, A., Shardlow, J., & Wampold, B. E. (2017).** Attention Is All You Need. *Advances in Neural Information Processing Systems, 30*, 5998–6008.

[8] **Huang, G., & He, R. (2020).** Natural Language Understanding: A Survey. *Journal of Computer Science and Technology, 35*(5), 73–99. doi:10.1007/s11390-020-0058-1.

[9] **Dataset source:** *This Project "**DEVELOPMENT AND PERFORMANCE EVALUATION OF AN APPLICATION FOR NEWS ARTICLE SUMMARIZATION, CLASSIFICATION, AND SENTIMENT ANALYSIS USING DEEP LEARNING MODELS**" uses a dataset of news articles from **The Star Malaysia**, a reputable news source covering various topics including nation, technology, crime, and more. It includes details such as article titles, text, authors, publication dates, keywords, and summaries. Link - https://www.kaggle.com/datasets/azraimohamad/news-article-weekly-updated*