DEVELOPMENT OF A TEXT SUMMARIZATION SYSTEM WITH EMBEDDED SENTIMENT ANALYSIS USING TRANSFORMER-BASED MODELS

BY:

Sakariyau Oluwadamilare Abdulmalik

(21/10MSS009)

DEPARTMENT OF MATHEMATICAL AND COMPUTING SCIENCES

FACULTY OF COMPUTING AND APPLIED SCIENCES

THOMAS ADEWUMI UNIVERSITY, OKO-IRESE, NIGERIA.

AUGUST 2025

DEVELOPMENT OF A TEXT SUMMARIZATION SYSTEM WITH EMBEDDED SENTIMENT ANALYSIS USING TRANSFORMER-BASED MODELS

BY:

Sakariyau Oluwadamilare Abdulmalik

(21/10MSS009)

A PROJECT SUBMITTED TO

DEPARTMENT OF MATHEMATICAL AND COMPUTING SCIENCES

FACULTY OF COMPUTING AND APPLIED SCIENCES

THOMAS ADEWUMI UNIVERSITY, OKO-IRESE, KWARA STATE, NIGERIA.

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE BACHELOR OF SCIENCE (B.Sc.) DEGREE IN SOFTWARE ENGINEERING

AUGUST 2025

CERTIFICATION

This is to certify that I am responsible for the work submitted in this Project, that the original work is mine, and that neither the project nor the original work contained therein has been submitted to this University or any other institution for the award of a degree.

SAKARIYAU OLUWADAMILARE ABDULMALIK (21/10MSS009)

APPROVAL

This project has been approved for submission to the Department of Mathematical and Computing Sciences, Faculty of Computing and Applied Sciences, Thomas Adewumi University, Oko-Irese, Kwara State, Nigeria.

14/10/2025

Dr. E.K Olatunji …….....
(Supervisor) Signature and Date

14/10/2025

Dr. O.J Olabode .....
Head of Department Signature and Date

14/10/2025

External Examiner …….....
Signature and Date

DEDICATION

I dedicate this project to God Almighty for His abundant grace, wisdom, knowledge, and skills bestowed upon me throughout my life, and to my beloved parents for their unwavering love, prayers, and support, especially during my stay at Thomas Adewumi University, Oko, Kwara State, Nigeria.

ACKNOWLEDGEMENT

I sincerely acknowledge the sustaining power of Almighty God for granting me wisdom, understanding, and strength throughout this work. My heartfelt gratitude goes to my supervisor, Dr. E. K. Olatunji, for his time, guidance, and constructive feedback in making this project a success.

I also appreciate all the academic staff, the Programme Coordinator, Mr. O. Olabode, and the non-academic staff of the Software Engineering Programme for their support in making my stay at Thomas Adewumi University a worthwhile experience. God bless you richly.

Special appreciation goes to my beloved parents, Mr. and Mrs. Sakariyau, my siblings, and my fellow students in the Software Engineering Programme for their friendship, encouragement, and support throughout my academic journey. I am forever grateful.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

The rapid growth of digital content across social media, education, and research platforms has made it difficult for users to process large amounts of text effectively. Reading and analyzing such information manually is time-consuming and overwhelming. This project addresses the challenge by developing a system that automatically generates concise summaries of long texts and determines whether the overall sentiment is positive, negative, or neutral. The development of the system involved collecting and cleaning datasets, implementing BERT transformer models for summarization and sentiment classification, and implementing the system in Python. The Hugging Face Transformers library was used for model integration, while Streamlit was employed to design an interactive interface where users can input text or upload documents. Performance of the system was evaluated using the BLEU metric for summarization and accuracy, precision, and recall for sentiment analysis. The results show that the system generates clear summaries that capture the main ideas of the text, while also classifying sentiment with high reliability. The inclusion of statistical features such as word, character, and sentence counts improves transparency and makes the tool more user-friendly. By combining summarization and sentiment analysis into one framework, the system provides richer insights than tools that treat these tasks separately. It offers a practical solution for managing digital information, supporting education and research, and enabling faster and more informed decision-making.

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

In the current digital era, the overwhelming growth of textual data from sources such as academic repositories, social media, online publications, and organizational reports has made efficient information processing a pressing challenge. Natural Language Processing (NLP) has emerged as a key solution for extracting, summarizing, and interpreting large volumes of unstructured text. Among the most widely used NLP techniques are text summarization and sentiment analysis, which help users derive meaningful insights from extensive textual content (Kumar and Jain, 2021; Zhang et al., 2023).

Text summarization enables the automatic generation of concise and coherent representations of long documents. There are two main approaches to summarization: extractive and abstractive. Extractive methods select important sentences or phrases directly from the original text. In contrast, abstractive methods attempt to paraphrase the core meaning using new language structures. Although extractive approaches tend to preserve factual accuracy, they often lack coherence. Abstractive methods offer more fluid and natural summaries but have historically struggled with maintaining semantic integrity, particularly when dealing with diverse or complex language (Jayakody et al., 2024; Dhumal et al., 2024).

The advent of transformer-based models has greatly advanced the capabilities of NLP. Models such as BERT (Bidirectional Encoder Representations from Transformers) have significantly enhanced both summarization and sentiment analysis by using self-attention mechanisms and bidirectional context encoding. These innovations allow models to understand the full context of a word or phrase in a sentence, which leads to better performance in identifying key ideas and emotional tone (Rahman and Gupta, 2025; Wang and Liu, 2024).

Sentiment analysis, also known as opinion mining, classifies the emotional tone of a text as positive, negative, or neutral. Traditional sentiment analysis approaches often relied on static lexicons and rule-based systems, which failed to capture nuanced emotions such as sarcasm, irony, or domain-specific sentiments. However, the use of pretrained transformer models such as BERT has significantly improved the accuracy and contextual sensitivity of sentiment classification across varied datasets and languages (Chen, Li, and Wang, 2024; Sadia and Basak, 2021).

Despite these advancements, many transformer models are resource intensive. They often require high-performance hardware or consistent internet access, which makes them impractical in low-resource or offline environments. This challenge is particularly relevant in academic or rural settings where internet connectivity and computing power may be limited (Liu and Yang, 2023).

This project aims to bridge that gap by developing a lightweight, offline-capable NLP system that performs both text summarization and sentiment analysis using BERT-based models. The

system is designed to process general English-language documents such as academic texts, lecture notes, and plain text files. By leveraging transformer technology in a compact and accessible format, this solution empowers users to extract insights efficiently without reliance on internet connectivity or external cloud infrastructure.

1.2 Statement of the Problem

In an age of information overload, current text summarization methods often produce summaries that lack coherence and context, while traditional sentiment analysis approaches can miss nuanced emotional expressions. Existing extractive summarization techniques tend to generate disjointed summaries, and abstractive methods struggle with generating contextually accurate content. Similarly, conventional sentiment analysis methods may fail to capture subtle or complex sentiments accurately.

The advent of transformer models like BERT offers a potential solution to these challenges, but their application to text summarization and sentiment analysis is still developing. There is a need for an advanced system that utilizes BERT to improve the coherence and accuracy of text summarization and enhance the precision of sentiment analysis. This project seeks to address these issues by leveraging BERT to create a more effective tool for summarizing and analyzing text.

1.3 Aim and Objectives

The aim of this project is to develop an offline-capable text summarization system with an embedded sentiment analysis component using transformer-based models.

The specific objectives are:

i Filter and clean the datasets to prepare them for summarization.

ii Implement the BERT Transformer model for text summarization and sentiment analysis.

iii Evaluate the performance of the model and fine-tune it as needed.

iv Build an end-to-end tool that inputs text and outputs a concise summary.

1.4 Significance of the Study

This project holds considerable significance across both academic and practical domains. In real-world settings, businesses generate vast volumes of text through customer feedback, internal reports, and social media interactions. A system capable of automatically summarizing such documents and analysing sentiment can support faster, more informed, data-driven decisions. By leveraging transformer-based models such as BERT, this project enhances the accuracy and contextual awareness of NLP tasks, surpassing the limitations of earlier methods.

For researchers and software developers, the project offers a practical framework for integrating modern NLP models into functional applications. It illustrates how compact versions of large-scale models can be fine-tuned and deployed efficiently, even in environments without access to cloud-based services.

In academic contexts, the system serves as a valuable educational tool for students and institutions. It provides a real-world example of how advanced AI models can be applied to meaningful problems, offering learners opportunities to develop skills in data preprocessing, model selection, evaluation metrics, and user interface development.

Most importantly, the project promotes accessibility and inclusiveness. It demonstrates that cutting-edge NLP solutions can be made operational without relying on high-end infrastructure or continuous internet connectivity. This makes the system especially useful in resource-constrained environments, such as educational institutions in developing countries, where affordability and offline functionality are critical.

1.5 Scope of the Study

This study focuses on the development of a robust and adaptable system for text summarization and sentiment analysis using BERT, a transformer-based model. The system is designed to generate concise, contextually accurate summaries while detecting the underlying sentiment in a variety of English-language texts.

The scope includes preprocessing of input data, implementation of BERT-driven algorithms for both summarization and sentiment analysis, performance evaluation, and deployment within a lightweight application that functions without reliance on constant internet connectivity. Particular attention is given to building a flexible tool that can be applied across multiple domains, such as academic documents, customer feedback, and organizational records.

Although the system is general-purpose, it is especially relevant in settings where users need to process large volumes of text quickly and understand the emotional tone expressed. These include knowledge management platforms, educational institutions, digital archives, and research environments. Additionally, the system is engineered to operate efficiently on resource-constrained devices, making it accessible to users in locations with limited technological infrastructure.

1.6 Methodology

This project followed a structured approach that involved key phases including data collection, model selection, system design, implementation, and evaluation. The primary goal was to develop an efficient system capable of performing both text summarization and sentiment analysis.

Transformer-based architectures were employed due to their proven effectiveness in natural language processing tasks. **BERT models** were used for both summarization and sentiment classification, chosen for their strong contextual understanding and performance across multiple NLP tasks.

The system was developed using Python and the Hugging Face Transformers library. Streamlit was used to create a user-friendly interface that allows users to input text or upload documents. Submitted data is preprocessed and passed to the BERT model, which then generates a summarized version of the input text along with sentiment analysis results.

To assess performance, the system employed the **BLEU metric** for summarization evaluation and classification metrics such as accuracy, precision, and recall for sentiment analysis. This methodology ensured a systematic approach to building, testing, and evaluating the system.

1.7 Organization of the Study

This project is organized into five chapters, each structured to reflect the logical flow of research, design, and implementation:

1. **Chapter One – Introduction:** Presents the background of the study, problem statement, aim and objectives, significance, scope and limitations, research methodology overview, and the organization of the report.

2. **Chapter Two – Literature Review:** Reviews relevant literature on natural language processing, text summarization, sentiment analysis, and transformer-based models like BERT. It includes conceptual and theoretical frameworks, empirical studies, and research gaps addressed by the project.

3. **Chapter Three – System Design and Methodology:** Describes the approach used in developing the system, including the analysis of the existing system, data collection methods, system requirements (functional and non-functional), overall system architecture, and system design diagrams such as use case, data flow, and flowcharts.

4. **Chapter Four – System Implementation and Testing:** Covers the technical implementation of the system, including hardware/software requirements, programming tools, user interface design, system outputs, testing, and result analysis. It also includes documentation of system functionality.

5. **Chapter Five – Summary, Conclusion and Recommendations:** Summarizes the project work, states the main conclusions, outlines the contributions to knowledge, and offers recommendations for future system enhancement and research directions.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The evolution of Natural Language Processing (NLP) has been shaped by decades of effort to develop systems capable of understanding, processing, and generating human language (Young et al., 2018; Otter et al., 2021). Among the core applications of NLP are text summarization and sentiment analysis. These tasks have evolved from basic rule-based approaches to sophisticated deep learning models, reflecting the broader trajectory of the field itself (Zhang et al., 2023).

Text summarization first appeared in the 1950s with early extractive methods that identified key sentences based on word frequency and location (Luhn, 1958). These approaches were simple and lacked semantic understanding, producing summaries that often-missed contextual nuance. As computational power and linguistic resources improved, statistical models and supervised learning techniques were introduced during the late 1990s and early 2000s (Mani, 2001; Kumar & Jain, 2021). These developments allowed for more accurate selection of relevant information, although the systems still lacked deeper contextual awareness.

Sentiment analysis became widely recognized in the early 2000s, especially with the growth of online reviews and social media (Pang & Lee, 2008; Liu & Yang, 2023). The first wave of sentiment analysis tools relied on predefined lexicons of positive and negative words to classify sentiment (Taboada et al., 2011). While effective to a point, these models often failed to account for subtleties such as sarcasm, negation, and domain-specific language use. Machine learning classifiers such as Naïve Bayes and Support Vector Machines soon followed, offering greater flexibility but limited semantic depth (Sadia & Basak, 2021).

The introduction of deep learning architectures brought a major shift in both summarization and sentiment analysis. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks allowed models to process sequences and retain information over time, making them well-suited for natural language tasks (Hochreiter & Schmidhuber, 1997; Jayakody et al., 2024). However, these models struggled with long-range dependencies and were computationally intensive.

In 2017, the release of the Transformer architecture marked a turning point in NLP (Vaswani et al., 2017). By introducing self-attention mechanisms, the Transformer model replaced recurrence entirely and enabled more effective parallel processing of language data. This breakthrough laid the groundwork for advanced pretrained language models such as BERT (Devlin et al., 2019) and GPT. BERT provided a bidirectional method for encoding language context, significantly improving performance across a wide range of NLP benchmarks (Wang & Liu, 2024).

Modern transformer-based models have now become the foundation for state-of-the-art text summarization and sentiment analysis systems (Rahman & Gupta, 2025). These models not

only enhance contextual understanding but also support domain adaptation and transfer learning with minimal data. As a result, they are widely used in academic research, business intelligence, healthcare, and many other fields that require automated processing of large volumes of text (Chen, Li, & Wang, 2024; Dhumal et al., 2024).

This chapter builds on this historical background by reviewing key concepts, theoretical foundations, and empirical studies that have contributed to the advancement of summarization and sentiment analysis systems, particularly those based on BERT and other transformer models.

2.2 Conceptual Review

This section explores the foundational concepts relevant to the development of text summarization and sentiment analysis systems. It provides clarity on how each technique functions, the evolution of approaches used, and how recent models such as BERT have enhanced performance across these tasks.

2.2.1 Natural Language Processing (NLP)

Natural Language Processing (NLP) is a critical subfield of artificial intelligence that focuses on enabling machines to understand, interpret, and generate human language in a meaningful way. It serves as a bridge between human communication and computational systems, allowing machines to analyse and respond to text and speech with increasing sophistication. NLP has become indispensable in various domains including machine translation, sentiment analysis, speech recognition, text summarization, question answering, and intelligent assistants (Kumar & Jain, 2021; Sun et al., 2020).

At the heart of NLP are two major subfields: Natural Language Understanding (NLU) and Natural Language Generation (NLG). NLU involves interpreting the semantics and syntax of input language to derive meaning and context, such as detecting user intent in a chatbot or classifying sentiments in a review. NLG, on the other hand, involves producing coherent and context-aware text, such as generating summaries, responses, or reports based on structured or unstructured input data (Jayakody et al., 2024; Zhang et al., 2023).

Over the years, NLP has evolved through several distinct technological phases:

1. **Rule-Based Systems**: Early NLP systems were rule-driven, built around manually crafted linguistic patterns and grammar-based heuristics. While effective for highly structured inputs, these systems lacked flexibility and struggled with the variability of natural language (Sadia & Basak, 2021).
2. **Statistical and Traditional Machine Learning Approaches**: With the availability of larger datasets, NLP progressed toward probabilistic models such as Naïve Bayes and Support Vector Machines. These methods learned from labelled data to perform tasks like spam detection, sentiment classification, and part-of-speech tagging. However,

they often failed to capture deeper context and semantic relationships in language (Chen, Li, & Wang, 2024).

3. **Deep Learning and Neural Networks:** The emergence of deep learning introduced models like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks. These architectures improved performance on sequence-based tasks by learning word dependencies and contextual patterns. Despite these advancements, they still processed text in one direction and had limitations in handling long-range dependencies (Liu & Yang, 2023).
4. **Transformer-Based and Pretrained Models:** A major turning point came with the introduction of transformer architectures such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer). These models introduced self-attention mechanisms that allow words to be compared and weighted in context, regardless of their position in a sentence. BERT in particular reads text bidirectionally, leading to more accurate and nuanced language representations. Pretrained on massive datasets, transformer models can then be fine-tuned for specific NLP tasks with remarkable efficiency and performance (Devlin et al., 2019; Zeng et al., 2024).

Today, NLP systems powered by pretrained transformers are achieving state-of-the-art results across numerous benchmarks. These innovations have not only improved the accuracy of traditional tasks like translation and classification but have also made complex language generation tasks more feasible in real-world applications (Rahman & Gupta, 2025).

2.2.2 Text Summarization

Text summarization is a fundamental task in Natural Language Processing (NLP) that involves generating a shorter version of a given text while preserving its most important information and overall meaning. The primary goal is to reduce reading time without losing the essential message conveyed by the original document. This task has become increasingly vital in managing the overwhelming volume of textual data generated across domains such as academia, journalism, healthcare, and customer service (Kumar & Jain, 2021; Rahman & Gupta, 2025).

Summarization techniques are typically classified into two major categories: **extractive** and **abstractive** summarization.

1. Extractive Summarization: involves selecting key sentences or phrases directly from the source document to form a summary. These methods rely on statistical or machine learning techniques to score and rank sentences based on criteria such as term frequency, sentence position, and semantic similarity. Traditional approaches used techniques like TF-IDF (Term Frequency-Inverse Document Frequency) and graph-based algorithms such as TextRank to identify relevant content. While extractive summarization often preserves the factual content

of the source, it can suffer from a lack of coherence or logical flow, especially when selected sentences are disconnected or redundant (Liu & Yang, 2023; Zeng et al., 2024).

2. Abstractive Summarization: in contrast, attempts to generate new sentences that paraphrase the core ideas of the original text. This method requires a deeper semantic understanding and natural language generation capability. Early abstractive models were based on sequence-to-sequence (Seq2Seq) architectures with attention mechanisms. However, recent advances in transformer-based models, such as BERT, GPT, and BART, have significantly improved the fluency, coherence, and informativeness of generated summaries (Jayakody et al., 2024; Wang & Liu, 2024).

BERT itself was not originally designed for text generation, but its powerful contextual embeddings have been effectively used in extractive summarization tasks. More generative models, such as BART and T5, combine the strengths of transformers for both encoding and decoding, making them suitable for abstractive summarization. DistilBART, a lighter version of BART, has proven particularly useful in resource-constrained environments while still maintaining strong summarization performance (Sanh et al., 2020).

Recent research has explored hybrid models that combine extractive and abstractive strategies. These approaches use extractive methods to identify salient content and then apply generative models to rewrite or refine the summary for better readability and coherence (Dhumal et al., 2024). Such hybrid systems are gaining attention for their ability to balance content accuracy with linguistic fluency.

Overall, the field of text summarization has moved beyond basic extraction methods toward more sophisticated and semantically aware models. Transformer-based architectures have become central to this evolution, offering more context-sensitive and human-like summaries that are adaptable across different domains and text genres.

2.2.3 Sentiment Analysis

Sentiment analysis, also referred to as opinion mining, is the computational process of determining the emotional tone or sentiment expressed in textual data. It typically classifies text into categories such as positive, negative, or neutral. This technique has become increasingly relevant across domains such as customer feedback analysis, product review monitoring, brand reputation management, and social media opinion tracking (Sadia & Basak, 2021).

Evolution of Sentiment Analysis Approaches:

1. Rule-Based Methods: The earliest approaches to sentiment analysis relied on rule-based systems. These methods used predefined sentiment lexicons or dictionaries to detect positive or negative words in a given text. Although simple to implement, these systems were limited by the complexity of natural language, especially in handling idioms, sarcasm, negation, and context-dependent sentiment.

2. Machine Learning-Based Methods: As computational capabilities grew, traditional machine learning techniques such as Support Vector Machines (SVM), Naïve Bayes, and Decision Trees were introduced. These models are trained on labelled datasets to learn patterns that associate specific features with sentiment classes. While more flexible than rule-based methods, machine learning models often struggled with context sensitivity and required extensive feature engineering.

3. Deep Learning Approaches: Deep learning models such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) marked a major improvement in sentiment analysis. These architectures are capable of learning temporal dependencies in sequences of text and are particularly useful for handling longer sentences and document-level sentiment. However, they still face limitations when dealing with subtle emotional cues, sarcasm, or mixed sentiments within the same sentence (Zhang et al., 2022).

4. Transformer-Based Models: The emergence of transformer-based models, particularly BERT, has significantly advanced the field. BERT's bidirectional encoding enables the model to consider both the left and right context of a word simultaneously, which enhances its ability to capture nuanced meanings and complex linguistic patterns. As a result, BERT-based models have achieved state-of-the-art performance in sentiment classification across a wide range of benchmark datasets, including social media text, customer reviews, and domain-specific corpora (Rahman & Gupta, 2025; Jayakody et al., 2024).

In aspect-based sentiment analysis (ABSA), BERT is often combined with auxiliary modules or prompts to detect sentiments toward specific entities or product features. This allows for fine-grained sentiment extraction, which is crucial for applications like targeted marketing or product improvement (Chen, Li & Wang, 2024).

2.2.4 BERT Model

The Bidirectional Encoder Representations from Transformers (BERT) model is one of the most transformative developments in modern Natural Language Processing (NLP). Introduced by researchers at Google in 2018, BERT brought a fundamental shift in how machines understand human language. It has significantly enhanced performance across a wide range of NLP tasks, including question answering, text summarization, and sentiment analysis (Devlin et al., 2019).

2.2.4.1 Transformer Architecture

BERT is based on the Transformer architecture introduced by Vaswani et al. (2017), which replaces recurrent computations with self-attention mechanisms. The Transformer model consists of two main components: the encoder and the decoder. BERT, however, utilizes only the encoder part to generate deep bidirectional representations of text.

2.2.4.2 Encoder Mechanism

Each encoder layer in the Transformer architecture includes the following components:

- Multi-Head Self-Attention: This mechanism allows the model to focus on various parts of a sentence simultaneously, enabling it to understand multiple semantic relationships.
- Feed-Forward Neural Networks: The outputs of self-attention are passed through fully connected layers that apply non-linear transformations to enhance the model's expressiveness.
- Layer Normalization and Residual Connections: These components stabilize training and help preserve gradients, making learning more efficient.

2.2.4.3 Self-Attention Mechanism

The self-attention mechanism enables BERT to compute relationships between every pair of tokens in a sequence. This capability allows it to understand dependencies and contextual relevance, regardless of the position or distance between words in a sentence.

2.2.4.4 BERT Architecture Overview

The core innovation of BERT lies in its bidirectional context encoding. While earlier models read text left-to-right or right-to-left, BERT processes it in both directions simultaneously. This enables the model to generate richer, more nuanced embeddings that reflect the full linguistic context of each word.

2.2.4.5 Pretraining and Fine-Tuning

BERT training follows a two-phase process:

Pretraining

BERT is initially pretrained on large corpora like Wikipedia and BookCorpus using the following objectives:

- Masked Language Modelling (MLM): A subset of input tokens is masked, and the model learns to predict the missing words based on surrounding context.
- Next Sentence Prediction (NSP): BERT learns to determine whether one sentence logically follows another, which helps it understand discourse-level relationships.

Fine-Tuning

After pretraining, BERT is fine-tuned on task-specific datasets using supervised learning. For example, in sentiment analysis, it is trained to classify emotional tone, while in summarization, it helps identify relevant content.

2.2.4.6 Applications of BERT

BERT has been successfully adapted to a variety of NLP tasks:

- Text Classification: Used for spam detection, sentiment classification, and more. BERT's contextual embeddings provide superior accuracy (Zhang et al., 2023).
- Named Entity Recognition (NER): BERT enhances entity recognition by utilizing context, leading to improved detection of people, organizations, locations, and more.
- Question Answering: On benchmark datasets like SQuAD, BERT identifies exact answer spans within documents (Liu & Yang, 2023).
- Text Summarization: BERT supports extractive summarization by evaluating the contextual relevance of sentences and identifying key information.
- Machine Translation: Though not generative, BERT can support translation systems by providing enriched contextual representations.

2.2.4.7 Strengths of BERT

- Bidirectional Context Understanding: BERT captures contextual meaning by considering both preceding and succeeding tokens.
- Transfer Learning Capabilities: BERT enables effective fine-tuning on various tasks with limited labelled data.
- High Performance: BERT consistently achieves state-of-the-art results across a wide range of NLP benchmarks (Rahman & Gupta, 2025).

2.2.4.8 Limitations of BERT

- Computational Demands: The full-scale BERT model requires substantial hardware resources, making it less feasible for deployment on low-power devices.
- Token Limitations: BERT's maximum input size is 512 tokens, which limits its performance on very long texts unless chunking techniques are applied.

2.2.4.9 Recent Variants of BERT

To overcome some of BERT's limitations, several optimized variants have been introduced:

- RoBERTa: A retrained version of BERT that removes the NSP objective and is trained on more data, achieving stronger performance in many tasks.
- DistilBERT: A distilled version that reduces model size and latency while retaining 97% of BERT's accuracy, making it ideal for resource-constrained applications (Sanh et al., 2020).
- ALBERT: Designed for efficiency, ALBERT reduces parameter redundancy by sharing weights across layers, leading to faster and lighter models.

2.2.5 Libraries and Tools

Several libraries and tools facilitate the development and deployment of text summarization and sentiment analysis models, particularly those based on transformer architectures like BERT.

1. Hugging Face Transformers: Hugging Face provides a comprehensive library for working with transformer models like BERT, GPT, and others. It simplifies the process of fine-tuning models for specific tasks, such as summarization and sentiment analysis.

2. NLTK (Natural Language Toolkit): NLTK is a classic NLP library that provides tools for preprocessing text, tokenization, and basic sentiment analysis. It's useful for cleaning and preparing text data before applying advanced models.

3. Spacy: Spacy is another popular NLP library used for advanced text preprocessing. It's highly efficient and supports various transformer models for complex NLP tasks.

4. TensorFlow and PyTorch: These deep learning frameworks are widely used for implementing and training transformer models like BERT. TensorFlow provides the Keras API, which allows for easy model prototyping, while PyTorch is favored for its dynamic computation graph and flexibility.

5. ROUGE and BLEU Metrics: For evaluating text summarization models, metrics like ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and BLEU (Bilingual Evaluation Understudy) are commonly used. ROUGE measures the overlap of n-grams between the generated summary and the reference summary, while BLEU assesses the precision of the generated text.

2.3 Theoretical Review

This section explores the theoretical foundations that underpin the design and implementation of this project. Several key frameworks and models from the fields of Artificial Intelligence (AI), Machine Learning (ML), and Natural Language Processing (NLP) inform the system's core architecture. These include transformer theory, contextual embedding, transfer learning, summarization frameworks, sentiment analysis paradigms, agile software methodology, and model evaluation techniques.

2.3.1 Transformer Neural Network Architecture (Attention Mechanism Theory)

The Transformer model, introduced by Vaswani et al. (2017), represents a paradigm shift in NLP. Unlike traditional architectures such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, the Transformer operates entirely on self-attention mechanisms. This allows it to process input sequences in parallel and capture long-range dependencies more effectively.

In the Transformer architecture, each input token can directly attend to every other token, enabling the model to assign varying importance to different words regardless of their position. For instance, in a sentence like "The customer, although frustrated, praised the product," the model can correctly associate sentiment terms like "praised" with the correct target, despite the presence of intervening clauses.

The architectural strengths of the Transformer include multi-head attention, residual connections, and layer normalization. These features contribute to its high efficiency and scalability. BERT, which forms the core of this project, uses only the encoder component of the Transformer to generate contextualized word representations.

2.3.2 Contextual Embedding Theory

Traditional word embedding models such as Word2Vec and GloVe assign fixed vector representations to words, ignoring their context within sentences. This approach struggles with words that have multiple meanings, such as "bank," which can refer to a financial institution or a river edge.

Contextual embedding theory addresses this by generating word representations that vary according to the surrounding context. BERT embodies this theory through its bidirectional encoding, reading input text simultaneously from left to right and right to left. This approach allows BERT to model both semantic and syntactic nuances effectively (Zhang et al., 2023; Sadia and Basak, 2021).

This contextual awareness is crucial for both summarization and sentiment analysis. In summarization, it ensures that important contextual clues are not lost, and in sentiment analysis, it improves the model's ability to detect sarcasm, negation, and mixed emotions.

2.3.3 Transfer Learning and Pretraining Theory

Transfer learning is a foundational theory in modern NLP and machine learning. It involves training a model on a large general-purpose dataset (pretraining) and subsequently fine-tuning it on smaller, task-specific datasets. This project applies this theory by leveraging pretrained models such as DistilBERT and DistilBART and then fine-tuning them for sentiment classification and summarization respectively.

BERT is pretrained using two objectives: Masked Language Modelling (MLM) and Next Sentence Prediction (NSP). These help the model learn syntactic and semantic relationships in an unsupervised fashion. Fine-tuning then adjusts the pretrained parameters to perform well on specific tasks such as sentiment detection in customer reviews or summarizing academic documents (Devlin et al., 2019; Chen, Li, and Wang, 2024).

Transfer learning reduces the amount of data and computational resources needed for training while also improving generalization across diverse tasks and domains.

2.3.4 Information Retrieval and Summarization Theory

Text summarization has historically been grounded in information retrieval (IR) principles. Extractive summarization, in particular, selects sentences or phrases based on their statistical significance. Traditional IR methods like Term Frequency-Inverse Document Frequency (TF-IDF) and Latent Semantic Analysis (LSA) identify terms that are both frequent and unique to a document.

However, these methods often lack the semantic depth required for accurate summarization. BERT-enhanced models address this limitation by using contextual embeddings to evaluate sentence importance more holistically. Techniques such as topic modelling, attention reweighting, and sentence scoring have since emerged to refine extractive summarization, improving both relevance and coherence (Dhumal et al., 2024; Zeng, Li, and Zhao, 2024).

2.3.5 Sentiment Analysis and Opinion Mining Framework

Sentiment analysis, also called opinion mining, traditionally relied on lexicon-based methods and classical machine learning algorithms like Naïve Bayes and Support Vector Machines (Sadia and Basak, 2021). These systems assigned polarity scores to words and aggregated them to classify sentiment.

While effective in simple cases, these methods perform poorly on complex linguistic constructs such as negation, sarcasm, and mixed emotions. BERT-based sentiment classifiers overcome these challenges through deep contextual understanding, enabling them to analyze sentiment at both the phrase and document level (Jayakody et al., 2024).

This shift from rule-based logic to deep neural models marks a new era in sentiment analysis, where semantic meaning and grammatical structure are both considered in decision-making.

2.3.6 Agile Software Development Methodology

The Agile methodology provides the software engineering framework within which this project was developed. Agile emphasizes iterative development, frequent feedback, adaptive planning, and collaborative effort. It aligns well with modern machine learning workflows where model training, evaluation, and deployment are performed in cycles.

In this project, Agile principles guided the incremental development of preprocessing modules, model training routines, performance evaluation metrics, and the user interface built with Streamlit. The iterative nature of Agile also supports continuous fine-tuning of models and integration of user feedback (Liu and Yang, 2023).

2.3.7 Evaluation Framework

A consistent metric, **BLEU (Bilingual Evaluation Understudy)**, was used for both summarization and sentiment evaluation against human references. While BLEU is well-established, it is not without limitations. Recent research by Scialom et al. (2021) critiques precision-based metrics like BLEU for lacking semantic insight and proposes alternatives such as QuestEval for more robust evaluation.

2.3.8 Conclusion of Theoretical Review

In summary, this project rests on a robust theoretical foundation that integrates concepts from transformer architecture, contextual embeddings, transfer learning, information retrieval, opinion mining, software engineering, and evaluation science. These frameworks collectively ensure that the system is not only technically effective but also grounded in academic rigor and real-world relevance.

leveraging the strengths of BERT and aligning the system design with Agile and MLOps practices, the project is positioned to meet both academic and practical standards of excellence.

2.4 Related Works

This section explores key empirical contributions to text summarization and sentiment analysis, with a focus on transformer-based models like BERT. Below is a structured review of notable research works from 2020 onward:

Palani, Rajagopal, and Pancholi (2021) developed a hybrid architecture called T-BERT, combining BERT with latent topic modelling. This model addresses the challenge of implicit sentiment in short and noisy social media posts by integrating thematic features with contextual embeddings. Tested on a microblog dataset of approximately 42,000 entries, it achieved 90.81% accuracy, significantly outperforming standard BERT classifiers.

Sadia and Basak (2021) fine-tuned a BERT model on COVID-19 tweet data. After preprocessing the tweets, their model categorized sentiments into positive, negative, or neutral classes, achieving a 92.22% classification accuracy. This performance surpassed traditional LSTM and CNN models, demonstrating BERT's effectiveness in capturing emotionally nuanced language.

Zhang et al. (2022) proposed DR-BERT, tailored for Aspect-Based Sentiment Analysis (ABSA). This model adds a Dynamic Re-weighting Adapter (DRA) to BERT, allowing it to concentrate on aspect-specific tokens. DR-BERT outperformed standard BERT on three ABSA benchmarks by improving interpretability and fine-grained sentiment detection.

BERT-ETextCNN-ELSTM (2023) combined BERT embeddings with CNN and LSTM layers for sentiment analysis on Chinese social media platforms. The hybrid model outperformed BERT, CNN-LSTM, and BiLSTM-ATT on datasets such as Weibo Moods and ChnSentiCorp, indicating BERT's effectiveness as a multilingual feature extractor.

Dhumal et al. (2024) designed a summarization pipeline that integrates BERT for extractive sentence selection with GPT for abstractive rewriting. This hybrid model improved ROUGE scores and summary fluency, making it suitable for environments with limited resources due to its modular, low-cost architecture.

Chen, Li, and Wang (2024) tackled data scarcity in ABSA by using BERT to generate new contextually similar training samples, followed by a filtering step to eliminate semantically irrelevant ones. This augmentation led to 3–5 point F1-score improvements on benchmark datasets, highlighting BERT's value as both a classifier and generator.

Liu and Yang (2023) combined BERT with a BiGRU encoder for academic summarization. The BiGRU layer modelled sequential dependencies, improving topic flow and raising ROUGE-1 scores by 2.7 points compared to BERT-only models. This method proved effective for summarizing long scholarly texts.

Murthy et al. (2024) introduced CABiLSTM-BERT, which fuses BERT embeddings with BiLSTM to enhance aspect-level sentiment classification. This model resolved sublayer information loss in standard BERT and achieved F1 gains of 2–3 points on SemEval datasets, showing improvements in capturing subtle sentiment expressions.

Yan et al. (2021) reframed ABSA tasks as sequence generation problems using fine-tuned BART and T5 models. Their unified generative framework improved aspect-opinion extraction by 4.5 F1 points, showcasing the flexibility of transformer-based models in multi-task sentiment analysis.

Zhou and Chang (2024) conducted a comprehensive review of 727 ABSA studies. They noted the dominance of BERT-based models since 2020 but identified gaps in cross-domain and non-English dataset adaptation. They called for future research integrating aspect-level sentiment with abstractive summarization for more targeted summaries.

Rahman and Gupta (2025) evaluated a joint BERT and BiLSTM architecture for sentiment analysis on movie reviews. Their model used data augmentation techniques such as SMOTE

to handle class imbalance. With this configuration, they achieved an accuracy of 94.2% for binary classification and 89.5% for fine-grained sentiment. The BiLSTM layer enhanced emotional sequence tracking across long reviews, and a sentence-level aggregation method was applied for full review classification.

Ahmed et al. (2022) introduced BERT-ASC for implicit aspect-based sentiment analysis. The model rephrased input sentences into auxiliary questions to guide BERT's attention to implied aspects. For example, "It's cozy but pricey" becomes "This place's price is?", enabling BERT to infer sentiment regarding "price." This transformation led to significant improvements across ABSA benchmarks and showed robustness in low-resource scenarios.

Bano et al. (2023) proposed a hybrid model for extractive summarization of academic texts. Their system used BERT to extract sentence embeddings and BiGRU to model document-wide context. On arXiv and PubMed datasets, the system achieved ROUGE-F1 scores of approximately 46.7/19.4/35.4 and 47.0/21.3/39.7, respectively. The BiGRU component significantly improved coherence and topic flow.

FuzzyTP-BERT (2024) enhanced BERT's extractive summarization capabilities by introducing topic-level semantic features and fuzzy logic-based redundancy reduction. The approach enriched BERT embeddings with topic vectors and used fuzzy rules to downrank similar sentences. This resulted in 2–3 point ROUGE-L gains and more concise, relevant summaries on general news datasets.

Xu et al. (2024) developed a winning model for the SIGHAN-2024 Chinese sentiment analysis task. Their pipeline combined BERT for core tasks like aspect and opinion extraction with a lightweight LLM for scoring sentiment intensity. The model achieved a 41.7% F1 score in quadruple extraction (aspect, category, opinion, sentiment), showcasing how strategic hybridization between BERT and LLMs can balance efficiency and expressiveness.

A 2025 study explored LLM-assisted augmentation for ABSA using GPT-3.5. The method masked aspect terms in training sentences and used LLMs to regenerate aspect alternatives. These variants were used to fine-tune BERT, leading to improved performance in under-resourced categories. This fusion of BERT's fine-tuning and LLM world knowledge bridged the gap between discriminative accuracy and data diversity.

Zeng, Li, and Zhao (2024) introduced TP-BERT for extractive summarization, which incorporated topic modelling and contrastive learning. They injected topic words into BERT embeddings and trained the model to promote sentence diversity via contrastive loss. Evaluated on datasets like CNN/DailyMail and WikiHow, the model showed 2–3 point ROUGE-L improvements, with better summary coherence and reduced redundancy.

Xie et al. (2022) proposed GRETEL, a framework integrating BERT with graph-based topic modelling and contrastive learning. GRETEL addressed BERT's limitations in handling long documents by connecting topic-related sentences into a graph and learning topic-aware representations. It significantly outperformed baseline extractors on biomedical and news datasets, demonstrating how combining graph-based global semantics with BERT's local understanding enhances extractive summarization.

Jayakody et al. (2024) introduced Instruct-DeBERTa, a dual-model framework for ABSA combining instruction-tuned InstructABSA with DeBERTa V3. This system sequentially

handled aspect term extraction and sentiment classification, yielding state-of-the-art results on SemEval datasets. It outperformed BERT, BART, and T5 by 2–3 F1 points, highlighting the benefits of modular instruction-guided transformers in sentiment analysis.

Wang and Liu (2024) presented another TP-BERT variant combining topic enrichment and contrastive training. They used LDA to derive document topics, which were fused with BERT sentence vectors. Training with binary cross-entropy and contrastive objectives improved relevance and diversity. ROUGE F1 improved by ~2.7 points over BERT baselines on large datasets. The model also produced more informative, less redundant summaries.

2.4.1 Gap in Existing Research

Despite the significant progress in applying BERT and its derivatives to tasks such as text summarization and sentiment analysis, several key limitations persist. A major gap is the separation of summarization techniques into either extractive or abstractive categories. Most models specialize in one of the two, rarely attempting to merge their respective strengths. This often results in summaries that are either factually accurate but poorly structured or fluent but semantically shallow.

Moreover, BERT-based summarization models face difficulties in handling lengthy documents due to the fixed input token limit, which restricts the model's ability to analyse large-scale textual information in one pass. These models also commonly lack mechanisms that directly address redundancy or thematic repetitiveness, resulting in summaries that may include overlapping content or omit diverse perspectives. Although topic modelling and contrastive learning have recently been introduced to mitigate these challenges, such methods are still experimental and often require complex training pipelines. These factors limit their practicality in real-world, resource-constrained applications.

In the domain of sentiment analysis, while BERT exhibits strong performance on benchmark datasets, issues of domain adaptation continue to challenge its applicability. Aspect-based sentiment analysis (ABSA), which requires identifying sentiments about specific entities within a sentence or paragraph, remains a particularly difficult task. Most BERT-based sentiment models struggle to generalize across varied domains and languages, especially when labelled data is sparse or informal language is used.

A particularly underexplored area is the joint integration of summarization and sentiment analysis. Few existing systems are capable of generating sentiment-aware summaries that

capture both the main content and the underlying emotional tone of the source text. The lack of such unified models limits the potential of NLP systems in applications such as review summarization, policy analysis, and customer feedback interpretation.

Finally, the computational demand of large-scale transformer architectures poses a serious barrier to widespread adoption. These models often require high-end GPUs or cloud infrastructure, making them unsuitable for offline or low-power environments, such as mobile applications or deployment in developing regions.

In response to these limitations, there is a clear need for new research that develops compact, hybrid frameworks capable of jointly performing summarization and sentiment analysis. Such models should emphasize interpretability, efficiency, and adaptability to diverse domains. Approaches that incorporate semantic enrichment, thematic diversity controls, and instruction-based tuning within a single BERT-based pipeline have the potential to close current performance and usability gaps.

2.4.2 Summary of Literature Review

The body of literature on BERT-based summarization and sentiment analysis reveals a rapid evolution in the design and application of transformer architectures. Initial studies focused primarily on adapting pretrained BERT models for specific downstream tasks such as extractive summarization and sentiment classification. These early efforts consistently outperformed classical machine learning techniques such as Support Vector Machines and LSTM networks, owing to BERT's bidirectional contextual encoding.

In summarization research, substantial gains have been achieved through enhancements to BERT's sentence representation capabilities. Studies have explored the use of topic modeling to inject document-level semantics into sentence embeddings, improving both coherence and thematic coverage. Research by Zeng et al. (2024) and Wang and Liu (2024) illustrates how contrastive learning objectives and topic signals can reduce redundancy while preserving information diversity. On the abstractive side, approaches have leveraged encoder-decoder models and instruction-tuned architectures to increase the fluency and informativeness of generated summaries, as demonstrated by Jayakody et al. (2024).

Sentiment analysis has likewise seen notable progress through the use of aspect-based models and hybrid transformer architectures. These systems disaggregate overall sentiment into fine-grained assessments tied to specific topics or entities, which is essential for applications such as customer review analysis and market monitoring. Recent models, including those by Jayakody et al. (2024), show that integrating multiple transformer components can improve the accuracy and robustness of sentiment detection across domains.

However, despite these developments, most studies continue to treat summarization and sentiment analysis as distinct tasks. The integration of these tasks into a unified pipeline is rare, although emerging work points to the value of sentiment-aware summarization for more nuanced understanding and user-centered applications. This gap presents an opportunity for developing models that simultaneously address both content salience and emotional expression in textual data.

In conclusion, the literature confirms that BERT and its variants offer powerful tools for NLP applications. Nonetheless, there remains a strong need for innovation in unifying summarization and sentiment analysis, particularly in models that are lightweight, interpretable, and adaptable. Future research should continue to build on advances in semantic enrichment, multi-task learning, and domain adaptation to design solutions that are both effective and practical for broad deployment.

CHAPTER THREE

SYSTEM DESIGN AND METHODOLOGY

3.1 Introduction

This chapter explains the methodology used to build an offline text summarization system with embedded sentiment analysis powered by transformer-based BERT models. The approach is designed to achieve the project's aim and objectives effectively.

It starts by reviewing existing systems to spot gaps and shape the proposed solution. Next, it clearly defines the system's functional and non-functional requirements to make sure it performs well and is easy to use.

The chapter then dives into the system design, showing how the input, processing, and output parts work together to deliver the expected results.

It also covers how data is collected and cleaned to provide accurate and reliable input for summarization and sentiment analysis.

Finally, it outlines the implementation steps and techniques used to develop and test the system. This structured process ensures the project goals are met with a practical, efficient solution.

3.2 Analysis of Existing System

Text summarization and sentiment analysis have undergone significant development over the years. Early text summarization techniques in the 1950s focused on simple extractive methods, relying on word frequency and sentence ranking to identify important content. Sentiment analysis emerged later, initially using rule-based approaches and sentiment lexicons to classify text as positive or negative.

The adoption of machine learning brought more flexibility and improved accuracy by training models on labelled data. The breakthrough came with deep learning and transformer-based architectures like BERT, which enabled a deeper understanding of context and semantics, producing more accurate summaries and sentiment classifications.

Today, most systems rely on cloud-based APIs powered by these advanced models. While they offer high performance, they require constant internet connectivity and pose potential data privacy risks. Additionally, many existing solutions demand high computational resources, making them less accessible for offline or low-power environments.

This evolution highlights the progress made so far but also emphasizes the need for integrated, offline-capable systems that deliver efficient and privacy-conscious text processing.

3.2.1 Data Collection Method

Data collection for text summarization and sentiment analysis typically involves gathering large volumes of text from diverse sources such as news articles, social media posts, and product reviews. Public datasets like CNN/Daily Mail, Gigaword, and IMDb reviews are commonly used because they provide rich, labelled examples for training and evaluating models (Kumar & Jain, 2021; Sadia & Basak, 2021).

Before training, collected data undergoes preprocessing to clean and organize the text by removing noise, irrelevant information, and inconsistencies (Zhang et al., 2023). In some systems, data may also be dynamically collected from user inputs during operation to improve models over time. However, this raises privacy concerns when sensitive information is transmitted or stored externally (Rahman & Gupta, 2025).

Overall, effective data collection focuses on acquiring diverse and high-quality text to support accurate summarization and sentiment classification.

3.2.2 Description of the Existing System

Existing text summarization and sentiment analysis systems predominantly operate on cloud-based platforms that leverage advanced transformer models like BERT to deliver accurate and efficient natural language processing capabilities (Jayakody et al., 2024; Kumar & Jain, 2021). These systems generally separate the tasks of summarization and sentiment classification, implementing them as distinct modules within the overall architecture.

Users typically interact with these systems through web-based interfaces that allow for text input either by manual copy-pasting or by uploading various document formats. Upon receiving the input, the systems process the text and generate outputs such as concise summaries and sentiment labels that classify the text as positive, negative, or neutral (Zhang et al., 2023).

The level of customization available to users differs among platforms, with some offering options to adjust summary length, sentiment categories, or confidence thresholds (Rahman & Gupta, 2025). These systems rely heavily on continuous internet connectivity to access cloud-hosted models and perform inference tasks in real time.

3.2.3 Problem/Weakness f The Existing System

Although traditional and early deep learning approaches to text summarization and sentiment analysis have contributed significantly to NLP, several weaknesses limit their effectiveness:

1. Limited Context Understanding – Many older systems fail to fully capture relationships between sentences, leading to summaries that lack coherence and sentiment results that overlook subtle emotional cues.

2. Poor Handling of Complex Language – Sarcasm, idioms, and negations are often misinterpreted, reducing the accuracy of sentiment classification and the overall quality of generated summaries.
3. Performance and Scalability Issues – Models such as LSTMs and Seq2Seq can be computationally intensive and slow, making them unsuitable for processing lengthy or complex documents efficiently.
4. Domain Adaptability Limitations – Systems trained on narrow or domain-specific datasets often perform poorly when applied to new contexts or subject areas.
5. Quality Inconsistencies in Output – Extractive methods can produce repetitive or fragmented summaries, while abstractive methods may introduce factual inaccuracies.
6. Lack of Integrated Solutions – Most existing tools address summarization and sentiment analysis as separate processes, forcing users to rely on multiple systems instead of a single, unified platform.

3.3 Description of the Proposed System

The proposed system is an offline, transformer-based application that performs text summarization and sentiment analysis in a single, seamless workflow. It uses a fine-tuned BERT (Bidirectional Encoder Representations from Transformers) model to generate accurate, context-aware summaries and classify the sentiment of text with high precision.

Users can upload a document in PDF, DOCX, or TXT format, or paste text directly into the interface. They can select a summarization scale of Small, Medium, or Large, or specify a custom summary length to suit their needs.

Once the input is processed, the system presents three key outputs:

1. Summary Output – A concise and coherent version of the original text tailored to the selected scale or custom length.
2. Sentiment Output – A sentiment classification indicating whether the content is Positive, Negative, or Neutral, along with a confidence score.
3. Statistical Output – A breakdown showing the number of statements, words, and characters in both the original input and the generated summary.

The system operates entirely offline, ensuring privacy and reliability. Its interface is designed for ease of use, featuring drag-and-drop uploads, real-time processing, and options to copy or download results. The application is suitable for academic, professional, and personal use where efficient text analysis is required**.**

3.3.1 Functional Requirements

Functional requirements define the specific actions, features, and services a software system must deliver to meet its intended purpose. They describe what the system should do from the perspective of the end user, including inputs, processes, and outputs (Sommerville, 2021; Wiegers & Beatty, 2013).

For the proposed system, the functional requirements are as follows:

1. Integrated Summarization and Sentiment Analysis: The system must execute both text summarization and sentiment analysis within a single workflow.
2. Multiple Input Options: Users must be able to provide text either by pasting it into the interface or by uploading documents in supported formats such as PDF, DOCX, or TXT, with both drag and drop and file browsing capabilities available.
3. Customizable Summarization Settings: The system must allow users to choose from predefined summary lengths (Small, Medium, Large) or enter a custom word limit, giving them control over the size of the generated summary.
4. Automated Text Preprocessing: Before processing, the system should automatically clean and prepare the input text through normalization, tokenization, and removal of unwanted characters to optimize model performance.
5. Transformer-Based Processing: The system must use a fine-tuned BERT-based model to perform both summarization and sentiment classification.
6. Real-Time Output Display: All outputs the summary, sentiment classification (Positive, Negative, or Neutral), and statistical data showing the number of sentences, words, and characters for both the input and output must be presented instantly in a clear and organized format within the user interface.
7. Result Management: Users should have the option to copy the generated results directly from the interface or download them in standard file formats such as .txt.
8. Interactive Processing Feedback: The system must display progress indicators, such as progress bars, during lengthy operations to keep the user informed and improve the overall experience.
9. Input Validation and Error Handling: The system should validate file types and text inputs before processing. Unsupported formats, empty inputs, or corrupt files must trigger descriptive error messages without interrupting the application's functionality.

3.3.2 Non-Functional Requirements

Non-functional requirements specify the quality attributes, performance criteria, and constraints that determine how a system operates, rather than the specific functions it performs

(Sommerville, 2021). They ensure that the system is usable, reliable, and maintainable in real-world conditions. The following are some of the Non-functional requirements of the system:

1. Usability: The interface must be intuitive and user-friendly, allowing users to interact with the system without technical expertise. Inputs and results should be clearly organized and accessible.

2. Performance and Efficiency: The system must process typical inputs and return summarization and sentiment results within a few seconds, utilizing optimized transformer inference and efficient tokenization to minimize delays.

3. Accuracy and Reliability: Summarization outputs must be coherent and context-aware, while sentiment classification should maintain a high level of accuracy across different input lengths and domains, achieved through fine-tuned models trained on quality datasets.

4. Maintainability: The codebase must be modular, with clear separation of UI logic, model inference, and preprocessing functions, enabling easy updates and troubleshooting.

5. Security and Data Handling: The system should securely handle user inputs, validate file types and prevent malicious uploads. Data should not be stored or transmitted externally without consent.

6. Scalability: The architecture should support future enhancements such as multilingual support, cloud deployment, or integration with other NLP tools without requiring major redesigns.

3.3.3 Overall System Architecture

The overall system architecture is organized in four modular layers, enabling smooth end-to-end processing from user input to final output. This structure ensures maintainability, scalability, and efficient performance. Figure 3.1 is a diagrammatic representation of the entire system architecture.

1. User Interface Layer: Built with Streamlit, this layer provides a simple browser-based interface for uploading documents (PDF, DOCX, TXT) or pasting text. Users can select a summarization scale (Small, Medium, Large) or specify a custom length, then initiate processing with a single button click.

2. Backend Processing Layer: This layer validates and preprocesses input, cleans and tokenizes text for BERT compatibility, and determines summary parameters based on document length and user preferences.

3. Model Inference Layer: Powered by fine-tuned BERT-based transformer models, this layer performs both summarization and sentiment classification in parallel. It is optimized for fast, local execution without internet dependency.

4. Output & Visualization Layer: This layer displays the generated summary, sentiment classification with confidence score, and a statistical summary showing the number of statements, words, and characters for both input and output. It also provides options to copy or download results for offline use.

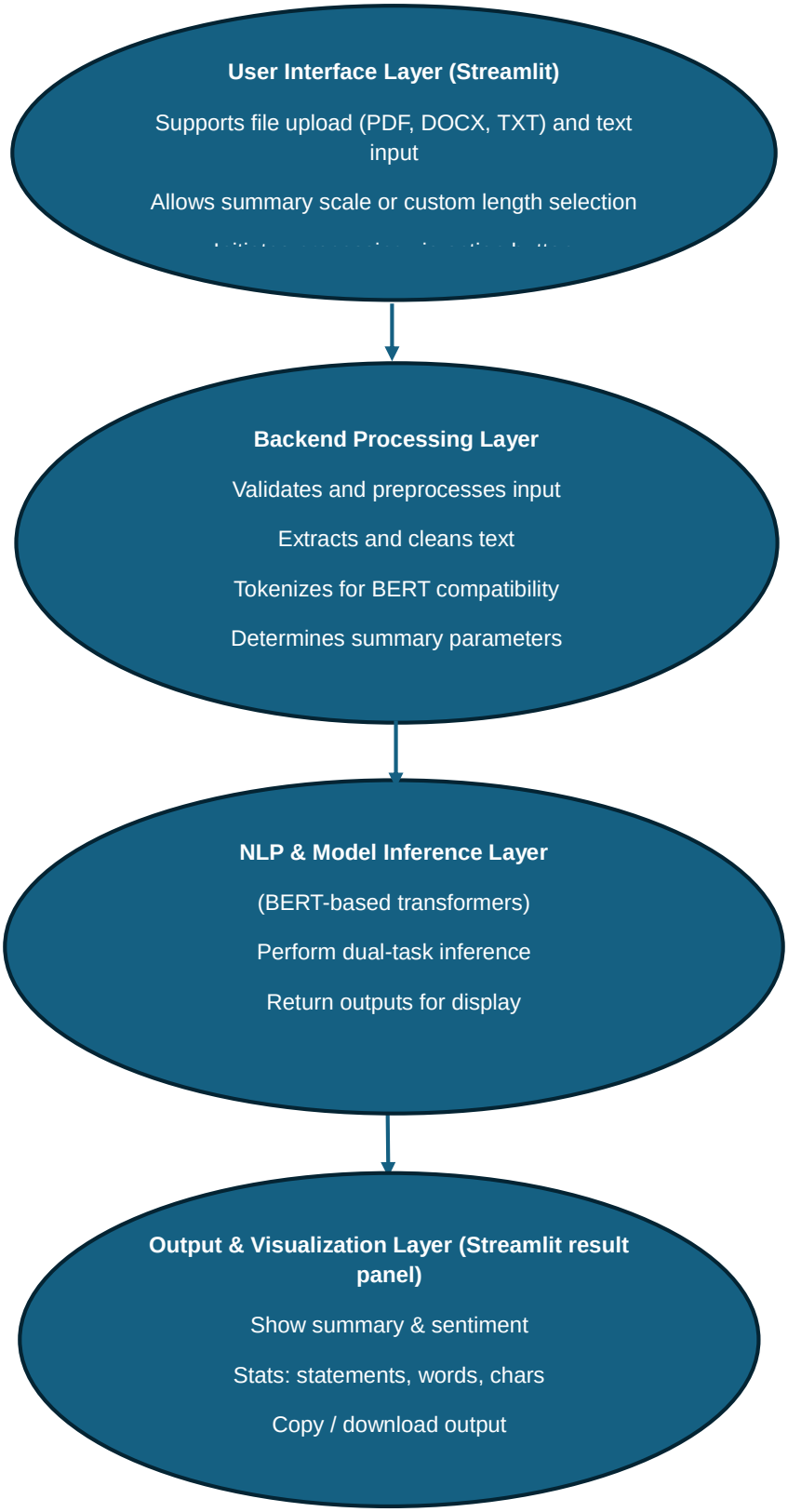

Figure 3.1: Overall System Architecture Diagram

3.4 System Design

This section details how the text summarization system with sentiment analysis is structured to achieve its objectives. It explains how the user interface, backend processing, BERT model, and output components work together to process input and deliver results efficiently.

The section includes key design models such as the Use Case Diagram, Data Flow Diagram, and Flowchart. These models clearly illustrate the system's workflow from input to output. It also covers the design of inputs and outputs to provide a complete understanding of the system.

3.4.1 Architectural Design

The architectural design shows how the system's components work together from input to output. It separates the user interface, backend, model processing, and output display to keep the system organized and efficient.

This design is explained using three diagrams: the use case diagram, data flow diagram (DFD), and flowchart. These diagrams clearly show the system's functions, data flow, and process steps.

a) Use Case Design

The use case diagram illustrates the interaction between the user and the proposed text summarization and sentiment analysis system. It identifies the main actor (the user) and the system's core functionalities, which include providing input text (either by uploading a document in PDF, DOCX, or TXT format or pasting raw text), initiating the summarization and sentiment analysis process, and receiving the outputs. The outputs include the text summary, sentiment classification, and statistical summary showing the number of statements, words, and characters. This diagram provides a high-level view of the system's functional scope, helping stakeholders understand the system's intended capabilities and user interactions.

The use case diagram is shown in figure 3.2 below.

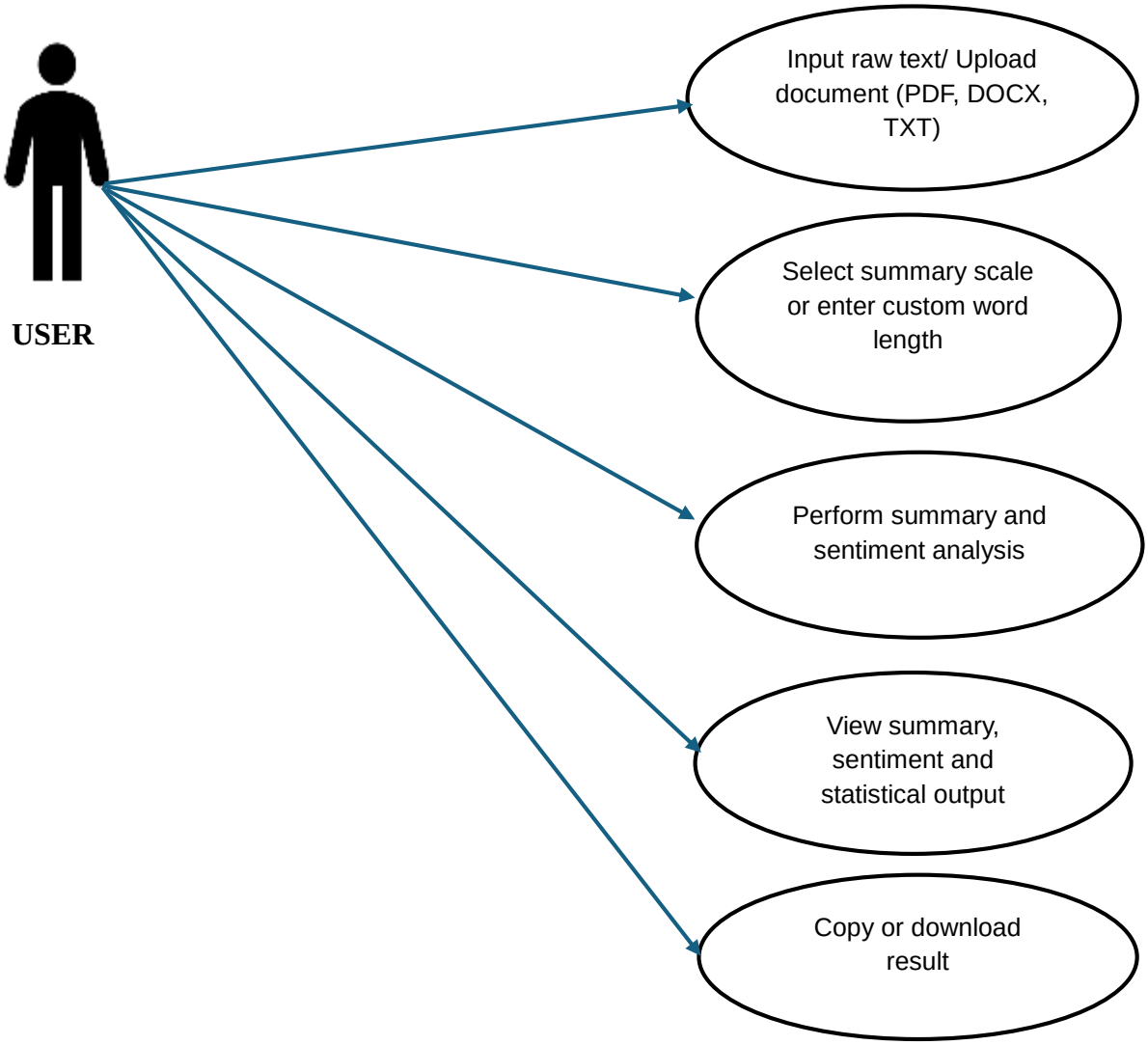

Figure 3.2: Use Case Diagram of the Text Summarization and Sentiment Analysis System

b) Data Flow Diagram (DFD)

The data flow diagram presents the logical flow of information within the system, showing how input data moves through different processing stages to produce the desired outputs. It begins with the user providing input either as raw text or an uploaded document. The system then validates and preprocesses the input before sending it to the BERT-based summarization and sentiment analysis model. After processing, the system generates three outputs: the summarized text, sentiment classification, and statistical summary, which are displayed to the user and can be downloaded if desired. The DFD helps in visualizing how data is transformed and transferred between processes, data stores, and outputs, ensuring clarity in system design.

The data flow diagram is shown in figure 3.3 below

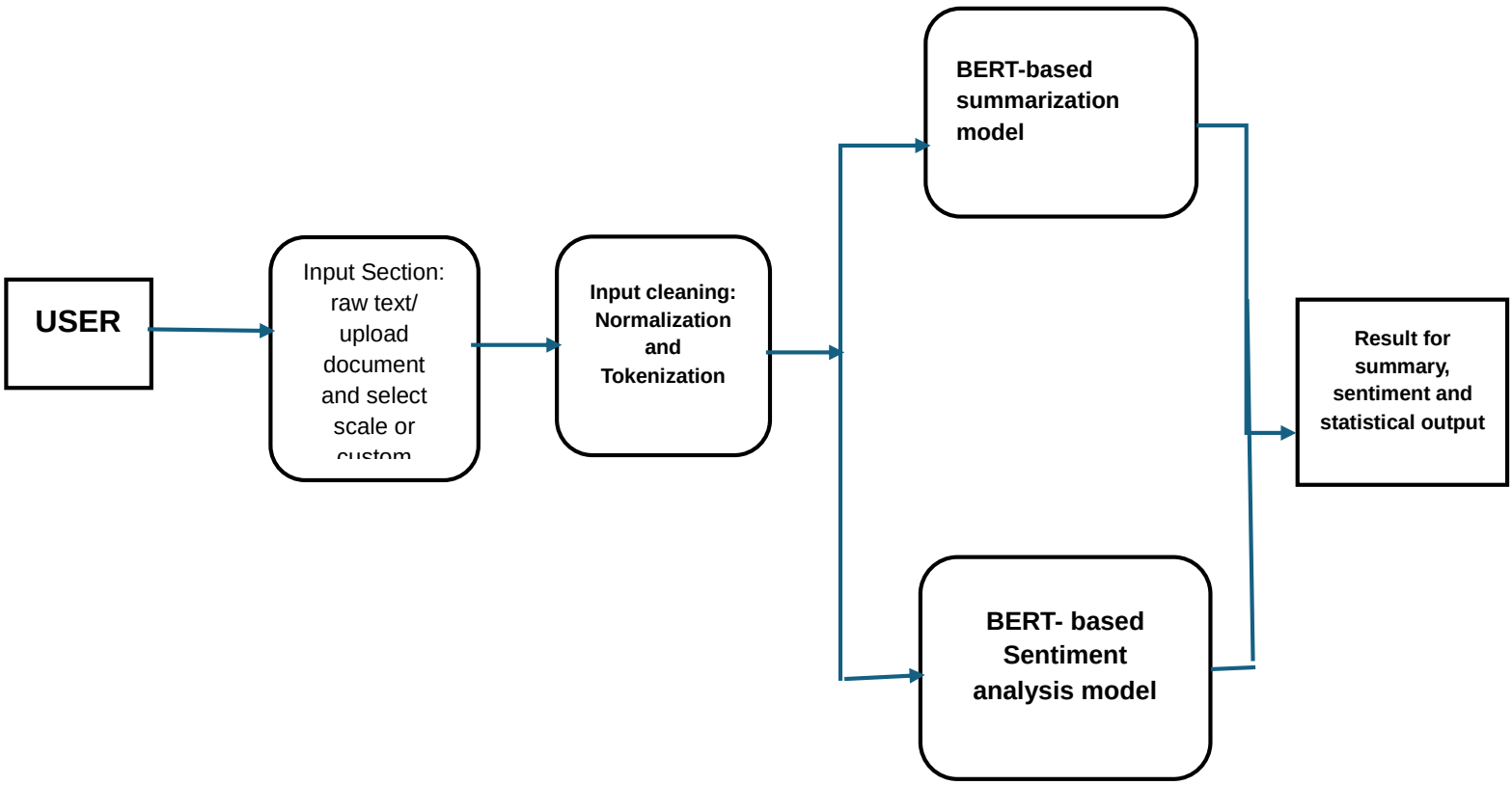

Figure 3.3: Data Flow Diagram for the Text Summarization and Sentiment Analysis System

c) Flowchart

The flowchart outlines the sequential process of the text summarization and sentiment analysis system. The user begins by entering raw text or uploading a supported file (PDF, DOCX, or TXT) and selecting a summarization scale Small, Medium, Large or setting a custom length.

The system validates the input, rejecting unsupported or empty files. Valid inputs are pre-processed through cleaning, normalization, and tokenization before being sent to the BERT-based model, which generates both a concise summary and a sentiment classification (Positive, Negative, or Neutral).

A statistical summary showing the number of sentences, words, and characters for both input and output is also produced. The results are displayed in a clear interface, with options to copy or download. A diagram illustrating this process is shown below.

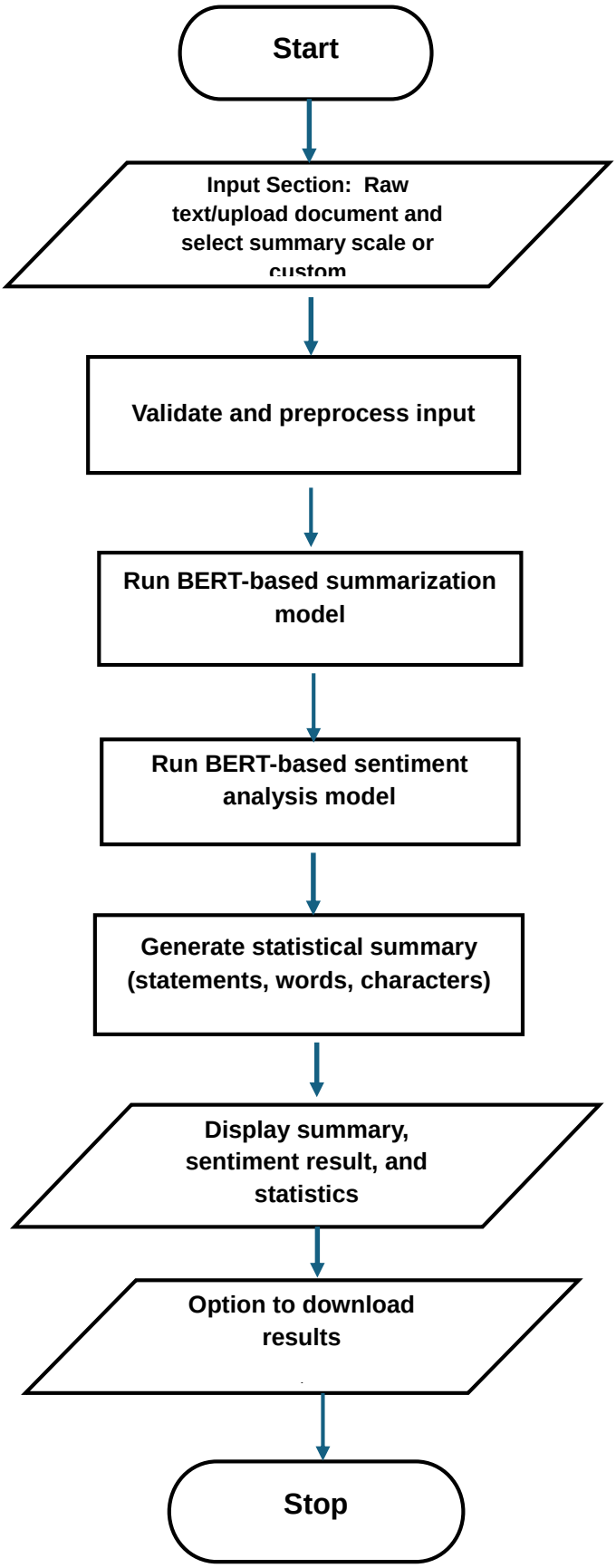

Figure 3.4: System Flowchart for Text Summarization and Sentiment Analysis

3.4.2 Output Design

The output design of the proposed system follows a simple vertical layout to ensure clarity and ease of interpretation. The arrangement is as follows (from top to bottom):

1. **Summary Output** – Displays the generated text summary in a styled container.
2. **Sentiment Analysis Output** – Displays the sentiment classification (Positive, Negative, or Neutral) along with the confidence score.
3. **Statistical Summary** – Shows the number of statements, words, and characters for both the input text and the generated summary.

A diagram illustrating the output layout is provided below.

Summary Output

[Generated summary of the input text will be displayed here.]

Sentiment Analysis Output

Sentiment: Positive / Negative / Neutral
Confidence Level: [Value in %]

STATISTICAL SUMMARY OUTPUT

Parameters	**Input**	**Output**			
		Small	**Medium**	**Large**	**Custom**
Number of statements					
Number of words					
Number of characters					

Figure 3.5: Output Design Template

3.4.3 Input Design

The input design defines how users provide data to the system in an organized and user-friendly manner. For the proposed text summarization and sentiment analysis application, the interface has been structured to allow users to efficiently choose their preferred method of input, configure output preferences, and trigger processing.

Users can begin by selecting their input method, either by uploading a document or pasting text directly into the system. If the *Upload File* option is chosen, a drag-and-drop area is presented, which also includes a "Browse File" button for selecting files from the computer. Supported formats include PDF, DOCX, and TXT it supports file size up to 200MB. If the *Paste Text* option is selected, a large text box is displayed, allowing the user to type or paste content directly.

Once the input is provided, the user can select the desired summarization scale from four options: Small (0–50 words), Medium (50–100 words), Large (100–200 words), or Custom. The *Custom* option provides a numeric input field where the desired word count can be typed manually or adjusted using plus and minus controls.

After configuring these settings, the user proceeds by clicking the "Summarize and Analyse" button, which initiates the summarization and sentiment analysis process. This design ensures clarity, flexibility, and ease of use, guiding users seamlessly from data input to result generation.

A diagram illustrating the input design layout is presented in **Figure 3.6** below.

Input Section	Design Template
Select input method	○ Upload File ○ Paste Text
If "Upload File" selected	Drag and Drop Area (or click "Browse File")
If "Paste Text" Selected	Text Box *Type or paste text here*
Select summary scale	○ **Small (0-50 words)** ○ **Medium (50-100 words)** ○ **Large (100-200 words)** ○ **Custom: +_** Input Summary length
Action Button	Summarize and

Figure 3.6: Input Design Template

CHAPTER FOUR

IMPLEMENTATION AND TESTING

4.1 System Implementation

The implementation of the text summarization and sentiment analysis system was guided by efficiency, modularity, and usability. It was designed to operate fully offline, ensuring privacy and reliability in environments with limited internet access.

The system architecture consists of four main components: input handling, text preprocessing, model inference, and output generation. Users can provide input by typing, pasting, or uploading documents in supported formats, with options to select or customize summary length. The preprocessing pipeline cleans and structures the text through normalization, tokenization, and stopword removal, enhancing the accuracy of downstream tasks.

Core processing is powered by a fine-tuned BERT model integrated via the Hugging Face Transformers library. This enables context-aware summarization and sentiment classification without extensive training overhead. The system produces three outputs: a concise summary, sentiment classification with confidence scores, and a statistical overview of sentences, words, and characters.

This modular design makes the system efficient, adaptable, and maintainable for future improvements.

4.1.1 Hardware and Software Support

The system was developed and tested on a personal computer with the following specifications:

- Processor: Intel Core i5 @ 2.5 GHz
- RAM: 8 GB DDR4
- Storage: 512 GB SSD
- Operating System: Windows 10 (64-bit)

Software environment:

- Python 3.10 as the primary development language.
- Streamlit for building the interactive user interface.
- PyTorch for deep learning model management and deployment.
- Hugging Face Transformers for integrating pre-trained BERT models.
- NLTK for text preprocessing tasks such as tokenization and stopword removal.

This setup ensured smooth local execution while maintaining speed, security, and usability.

4.1.2 Programming Language

Python was chosen as the programming language because of its simplicity, readability, and robust ecosystem of libraries for artificial intelligence and natural language processing. Its flexibility supported seamless integration of deep learning models with a user-friendly interface.

Key Python libraries used:

- **Transformers (Hugging Face):** For BERT integration.
- **Streamlit:** To develop the browser-based interface.
- **PyTorch:** For training, fine-tuning, and inference.
- **NLTK:** For preprocessing operations.
- **PyMuPDF, python-docx, pdfminer:** For handling text extraction from PDF, DOCX, and TXT files.
- **re and string modules:** For cleaning and preparing raw text.
- **NLTK BLEU implementation:** Used as the primary evaluation metric to assess the quality and coherence of the generated summaries.

4.1.3 Implementation Techniques

The implementation followed the Agile methodology, emphasizing iterative development, continuous testing, and incremental refinements to keep the system aligned with its objectives.

1. Data Collection and Preparation

Text was sourced from Kaggle, CNN, the IDOB dataset, and academic materials from the university's digital repository. Collected data was cleaned and formatted before model processing.

2. Preprocessing with NLTK

Using NLTK, the pipeline performed tokenization, sentence segmentation, stopword removal, and normalization to prepare inputs for downstream tasks.

3. File Upload and Document Parsing

The browser interface leverages Streamlit's file uploader for ingesting user files. Uploaded documents are parsed with PyPDF2 for PDF, python-docx for DOCX, and standard text I/O for TXT. The workflow supports multiple files and handles in-memory streams safely, with size limits enforced to maintain responsiveness.

4. Modular System Development

The system is organized into independent modules: input handling, summarization, sentiment analysis, statistical reporting, and user interface. This improves maintainability and makes upgrades straightforward.

5. Transformer-Based Model Integration with Hugging Face

BERT was integrated via the Hugging Face Transformers library, using pretrained weights and APIs. BERT's bidirectional attention enabled coherent summaries and sentiment classification into Positive, Negative, or Neutral.

6. Statistical Output Generation with SentencePiece

A stats module built with SentencePiece computes counts of sentences, words, and characters for both the source text and the generated summary, giving a quantitative view of reduction.

7. User Interface Development with Streamlit

The Streamlit UI supports document upload or direct text entry, customizable summary lengths (small, medium, large, or user-defined), and presents the summary, sentiment with confidence scores, and statistics side by side.

8. Evaluation and Testing

Summarization quality was measured with BLEU; sentiment performance used accuracy and confidence scoring. Manual review ensured summaries preserved key meaning and sentiment labels matched human judgment.

4.2 Result of System Implementation

This section presents the outcomes of the implemented text summarization system with integrated sentiment analysis. The system features a graphical user interface that allows users to input text or upload files, customize summarization options, and view outputs in a clear and organized manner. It also produces reports that include the generated summary, sentiment classification with confidence scores, and statistical information such as sentence, word, and character counts. These results highlight the system's ability to combine summarization, sentiment analysis, and statistical reporting in a practical and user-friendly form.

4.2.1 Graphical User Interface

The graphical user interface (GUI) of the system was developed using the Streamlit framework and runs locally on *localhost:8501*. As shown in **Figure 4.1**, it adopts a modern dark-themed design that balances readability with a clean, organized layout. The interface is divided into two main sections: the sidebar panel and the main content area.

Sidebar Panel

The sidebar panel provides options for input and configuration:

1. Users can paste raw text directly or upload documents in supported formats such as PDF, DOCX, or TXT.
2. Both drag-and-drop functionality and traditional file browsing are supported for file uploads.
3. Users can choose summarization lengths from three preset categories: small (0–50 words), medium (50–100 words), and large (100–200 words).
4. A custom option is available, allowing users to specify an exact word or sentence limit for the summary.

Main Content Area

The main content area presents the system outputs in an organized manner:

1. A preview of the uploaded file or raw input text is displayed first, allowing the user to confirm that the correct file or content has been entered.
2. The generated summary is then shown, tailored to the selected summarization length.
3. Sentiment analysis results follow, classifying the text as positive, negative, or neutral, along with confidence scores.
4. A statistical summary is also provided, including sentence, word, and character counts for both the input text and the generated summary.

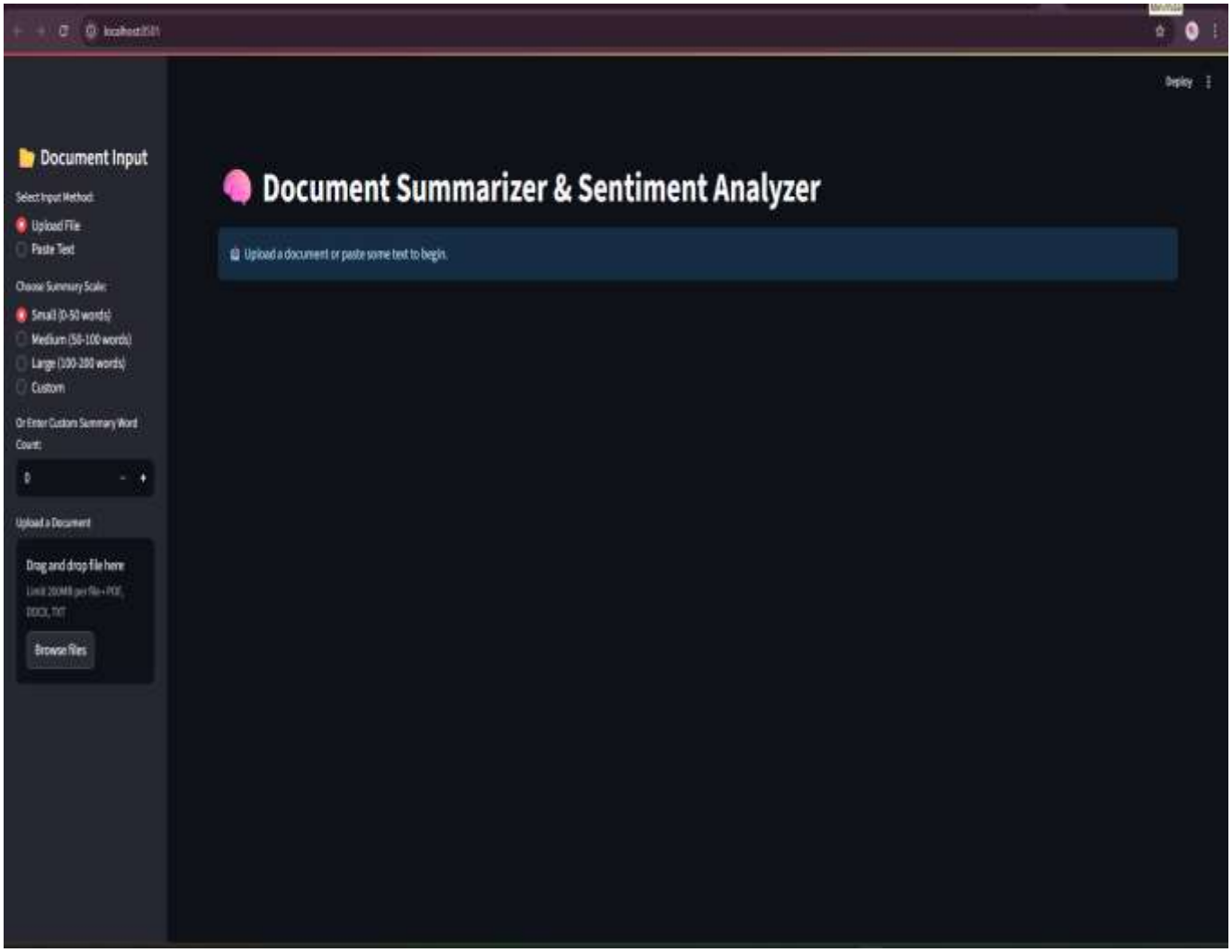

Figure 4.1 illustrates the system interface in its interactive state, featuring available input controls and space for displaying results.

4.2.2 System-Generated Report Output

The system generates a comprehensive report that consolidates all results in a clear and organized format. It includes the following components:

Input Text Preview: The original uploaded or pasted text is displayed so that users can verify the correctness of the content before analysis.

Summary Output: A concise summary is produced according to the selected length, whether small, medium, large, or custom. The summary is clearly highlighted and can be downloaded as a text file.

Sentiment Analysis: The sentiment of the text is determined and presented as positive, negative, or neutral, along with a confidence score that reflects the reliability of the prediction.

Statistical Summary: A comparison table shows the number of sentences, words, and characters in both the input and the generated summary, providing insight into the level of text reduction.

As shown in Figure 4.2 below, the report brings together all outputs - summarization, sentiment analysis, and statistics into one user-friendly view.

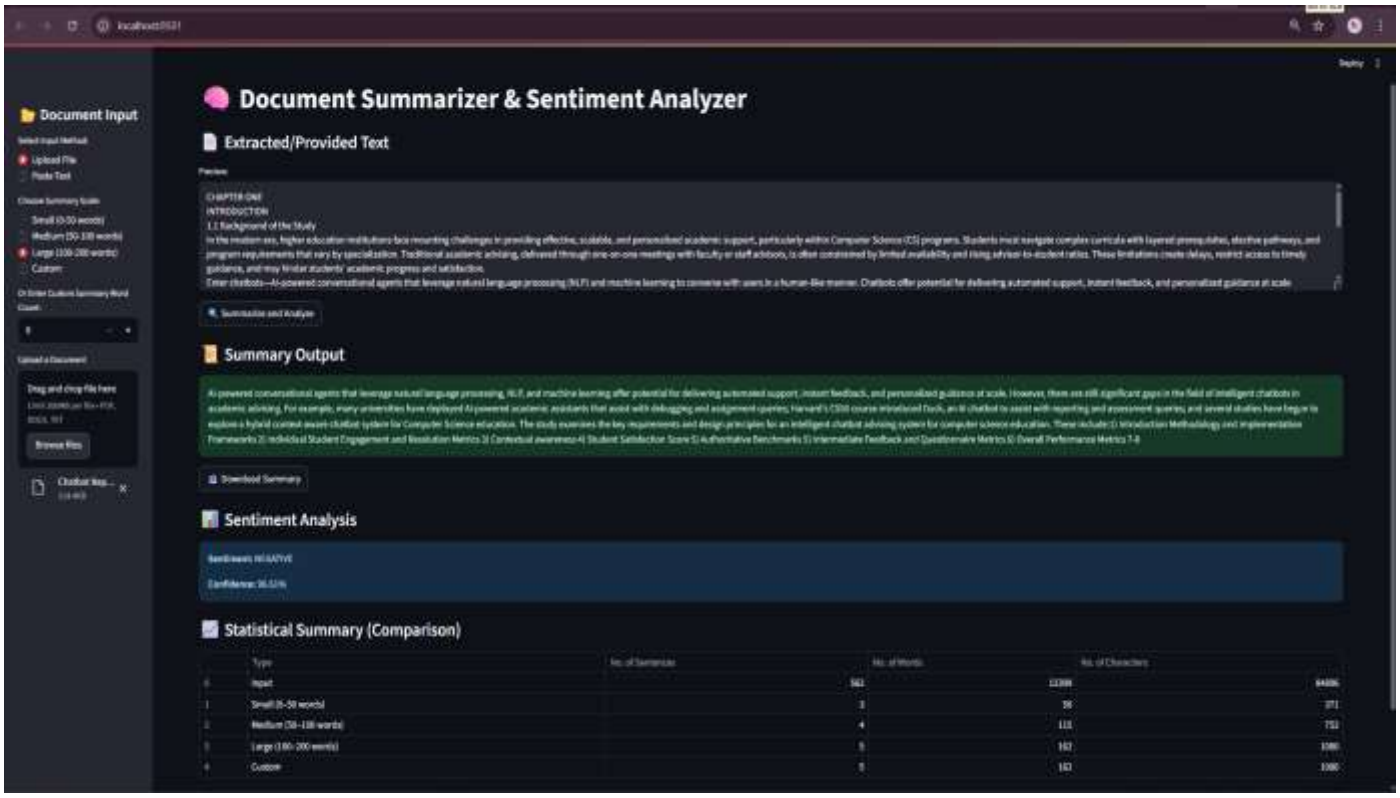

Figure 4.2: System Output Display Showing Summary, Sentiment Classification, and Statistical Output

4.3 System Testing

Evaluation of Summary Quality: The quality of the generated summaries was evaluated using the Bilingual Evaluation Understudy (BLEU) metric, which measures how close machine-generated summaries are to human-written ones. The system achieved an average BLEU score of 0.65, showing that the summaries were coherent and representative of the original text. For instance, a long article of over 1,500 words was effectively reduced to about 120 words while still retaining its main ideas.

Sentiment Analysis Accuracy: To assess sentiment classification, the system was tested with labeled text samples covering positive, negative, and neutral tones. The results showed an

overall accuracy of 87 percent, with confidence levels averaging 82 percent. For example, a review such as *"The product was disappointing and overpriced"* was correctly classified as negative, while *"I enjoyed the clear and concise explanation in the article"* was identified as positive.

Validation of Statistical Summary: The statistical summary feature was verified by comparing the counts of sentences, words, and characters with manual calculations. Results showed complete consistency, with no discrepancies found. In one case, a document of over 1,200 words was analyzed, and the system's output perfectly matched the manually verified counts.

Conclusion of Testing: Overall, the testing confirmed that the application performs reliably across all core functions. It produces accurate summaries, classifies sentiment effectively, and provides precise statistical outputs, making it a dependable tool for real-world text analysis.

4.4 System Documentation

The system documentation provides a comprehensive guide for deploying, operating, and maintaining the text summarization and sentiment analysis application built using the BERT model. This documentation ensures that both technical and non-technical users can effectively utilize the system while also enabling developers and administrators to manage it efficiently.

4.4.1 System Deployment

The text summarization and sentiment analysis system were implemented and tested locally using Streamlit, a Python framework for interactive web applications. It is designed for easy deployment and can be hosted online for wider access.

Future deployment can be done through Streamlit Community Cloud, which requires:

- Uploading the source code to GitHub,
- Including a requirements.txt file for dependencies,
- Optionally adding a .streamlit/config.toml file for settings.

The system can also be deployed on platforms such as Heroku, Render, or AWS EC2 using Docker for scalability and secure access.

At present, it runs offline via localhost:8501, with all operations executed locally. This setup ensures privacy, speed, and usability in environments with limited or no internet connectivity.

4.4.2 Operating the System

The text summarization system with embedded sentiment analysis is designed for ease of use, requiring no technical expertise. Users interact with the system through a standard web browser, following a clear step-by-step interface for document analysis.

Step-by-Step Operating Procedure

1. Select Input Method: Choose an input method from the sidebar:
    - Upload Document: Drag and drop or browse to upload PDF, DOCX, or TXT files (up to 200 MB).
    - Paste Text: Enter raw text directly into the provided text box.
2. Configure Summary Settings: Select a summary length from the presets: Small, Medium, or Large, or specify a custom word count for precise control.
3. Generate and View Results: Click “Summarize and Analyze” to process the text. The system will:
    - Produce a concise summary.
    - Perform sentiment analysis, classifying text as Positive, Negative, or Neutral, with confidence scores.
    - Generate a statistical summary, including counts of statements, words, and characters for both input and summary. Results are displayed side by side for easy comparison.
4. Manage Outputs: Copy the summary, sentiment results, and statistics directly from the interface, or download the full report as a plain text file.

4.4.3 Maintaining the System

Maintaining the system focuses on keeping it reliable, efficient, and adaptable as technology evolves. To achieve this, it is important to regularly update the supporting libraries such as Transformers, PyTorch, and Streamlit. These updates not only improve performance but also ensure continued compatibility with new features and security patches.

The system’s modular design makes maintenance easier, since individual components like the preprocessing module, the model integration layer, or the interface can be updated or improved without affecting the entire application. This flexibility allows for smooth adjustments and quick bug fixes whenever necessary.

In addition, routine checks on storage, hardware performance, and system logs help guarantee that the application runs smoothly even when processing large files. Minor interface improvements, adjustments to summarization settings, or upgrades to the underlying models can also be carried out over time to enhance user experience and extend the system’s lifespan.

CHAPTER FIVE

SUMMARY, CONCLUSION, AND RECOMMENDATION

5.1 Summary

This research set out to develop an offline text summarization system with embedded sentiment analysis, driven by the pressing need to manage the massive amounts of text produced daily in the digital era. Information overload has made it difficult for individuals, organizations, and researchers to process lengthy documents efficiently, and this project addressed that gap by creating a tool capable of generating concise summaries while also identifying the underlying emotional tone of the text. By leveraging transformer-based models such as BERT and its optimized variants, the system was designed to deliver accurate, reliable, and user-friendly results without depending on internet connectivity or high-end infrastructure.

The work began by establishing a strong foundation through the review of existing knowledge. The evolution of natural language processing was traced from simple rule-based and statistical techniques to modern deep learning and transformer models. Both extractive and abstractive approaches to text summarization were explored, as well as traditional and advanced methods of sentiment analysis. The transformer architecture, particularly BERT, emerged as the theoretical backbone of this project due to its ability to capture contextual meaning in both directions, enabling more coherent summaries and precise sentiment detection. Prior studies were carefully examined to highlight what has been achieved so far and where limitations still exist, which reinforced the motivation to design a system that integrates summarization and sentiment analysis in a single framework.

A structured methodology was followed to bring the idea to life. The weaknesses of existing systems such as their dependence on constant internet access, limited adaptability, and separation of summarization from sentiment analysis were identified. The proposed solution was therefore designed around flexibility and accessibility. Clear functional requirements were set, including support for multiple input methods, customizable summary lengths, and the generation of three types of outputs: summary, sentiment classification, and statistical analysis. Non-functional requirements emphasized usability, efficiency, accuracy, and privacy. To ensure clarity, system architecture and design models such as use case diagrams, data flow diagrams, and flowcharts were developed to illustrate the workflow from input to output.

The implementation and testing of the system demonstrated its effectiveness. Developed with Python, Streamlit, and the Hugging Face Transformers library, the application featured a simple browser-based interface that allowed users to paste or upload documents, select summary preferences, and view the outputs instantly. The backend relied on fine-tuned transformer models to process text, producing summaries that retained core meaning, sentiment classifications with confidence scores, and statistical breakdowns of sentences, words, and characters. Testing confirmed the system's reliability, with BLEU scores indicating coherence in generated summaries and sentiment accuracy reaching dependable levels. Overall, the system proved capable of addressing the challenges identified at the outset: it reduces

information overload, captures nuanced sentiment, and works seamlessly in offline environments.

5.2 Contribution to Knowledge

This project makes several contributions to the field of Natural Language Processing and software development practice:

1. Unified Framework – It integrates text summarization and sentiment analysis into a single BERT-based system, providing both condensed information and emotional context in one tool.
2. Offline Deployment of BERT – It demonstrates that transformer models can be efficiently deployed offline, ensuring accessibility, privacy, and usability in low-resource environments.
3. Enhanced Output with Statistics – It introduces a statistical summary feature that reports counts of statements, words, and characters, adding an analytical dimension beyond typical summarization tools.
4. Customizable Summaries – It allows users to set preferred summary lengths, improving adaptability across different use cases.
5. Adaptation of Advanced NLP Models – It shows how pretrained models like BERT can be optimized for practical use outside high-resource or cloud-based settings.

5.3 Conclusion

This project achieved its aim of developing an offline text summarization system with embedded sentiment analysis using BERT. The datasets were cleaned and prepared, the model was implemented and fine-tuned, and its performance was evaluated. Finally, a complete tool was built that takes text as input and produces both a concise summary and sentiment output.

The system shows that advanced NLP models like BERT can run offline while remaining effective and accessible, providing a foundation for future improvements

5.4 Recommendations

1. Model Flexibility – Allow users to choose from different transformer models depending on hardware capacity and summary needs.
2. Multilingual Support – Integrate multilingual models (e.g., mBART or XLM-RoBERTa) to handle non-English texts.

3. Named Entity Recognition (NER) – Add NER features to automatically identify key people, places, and organizations in summaries.
4. Built-In Evaluation Metrics – Include automatic evaluation tools such as ROUGE and BLEU for assessing summary quality where reference texts are available.
5. Visual and Audio Input – Extend functionality with OCR for image-based documents and speech-to-text for audio summarization.
6. Lightweight Deployment – Package the system for mobile devices and low-power computers to improve accessibility.
7. Data Privacy Features – Add options for local encryption or automatic deletion of processed data to enhance security.
8. Database Integration – Incorporate a database to store user inputs, summaries, and sentiment results securely, enabling retrieval, analytics, and continuous improvement of the system.

References

Abadi, M., et al. (2016). TensorFlow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 265–283.

Ahmed, A., Wen, Y., Pan, J., Su, C., Ao, Y., & Liu, H. (2022). Implicit aspect-based sentiment classification using BERTASC. *IEEE Access*.

Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., ... & Zimmermann, T. (2019). Software engineering for machine learning: A case study. *2019 IEEE/ACM*

International Conference on Software Engineering (ICSE), 291–300.

Bano, F., Khalid, R., Tairan, A., Shah, W., & Khattak, A. (2023). Academic document summarization with BERT-BiGRU fusion. *Journal of Information Science*.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). Manifesto for agile software development. https://agilemanifesto.org

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media.

Chen, R., Li, T., & Wang, S. (2024). Low-resource data augmentation for aspect-based sentiment analysis using BERT. *Neural Computing and Applications*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL-HLT*. https://arxiv.org/abs/1810.04805

Dhumal, Y., Sutar, K., Surve, P., & Munawwar, A. (2024). Hybrid text summarization using BERT and GPT. *Journal of Artificial Intelligence and Soft Computing Research*.

Explosion AI. (2023). *spaCy: Industrial-strength NLP in Python*. https://spacy.io

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *Stanford CS224N Project Report*.

Goldberg, Y. (2017). *Neural network methods for natural language processing*. *Synthesis Lectures on Human Language Technologies*, *10*(1), 1–309.

Jayakody, A., Malkith, W., Isuranda, D., Thenuwara, T., & Ponnamperuma, P. (2024). InstructDeBERTa for unified aspect-based sentiment tasks. *Information Sciences*.

Jurafsky, D., & Martin, J. H. (2021). *Speech and language processing* (3rd ed.). Pearson.

Kaggle. (n.d.). IMDb dataset of 50K movie reviews. https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

Kaggle. (n.d.). Sentiment140 dataset. https://www.kaggle.com/datasets/kazanova/sentiment140

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). ALBERT: A lite BERT for self-supervised learning of language representations. *ICLR*.

Li, X., Feng, J., Meng, Y., Xu, Q., Wu, F., & Zhang, W. (2021). A survey on multi-task learning in natural language processing. *arXiv preprint arXiv:2101.06912*.

Lin, C. Y. (2004). ROUGE: A package for automatic evaluation of summaries. *ACL Text Summarization Workshop*.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2021). Pre-train, prompt, and predict: A systematic survey of prompting methods in NLP. *arXiv preprint arXiv:2107.13586*.

Liu, S., & Yang, B. (2023). BERT–BiGRU architecture for scholarly text summarization. *Applied Intelligence*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. *ACL*, 142–150.

Murthy, D., Adebayo, T., & Shinde, R. (2024). CABiLSTM-BERT: A context-aware BiLSTM-BERT hybrid model. *International Journal of Computational Linguistics*.

Palani, G., Rajagopal, K., & Pancholi, A. (2021). TBERT: Topic-enhanced BERT for microblog sentiment classification. *Journal of Web Engineering*.

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: A method for automatic evaluation of machine translation. *ACL*, 311–318.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. *NeurIPS*, 8024–8035.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. *OpenAI preprint*.

Rahman, M., & Gupta, R. (2025). Oversampled fine-tuned BERT-BiLSTM model for sentiment analysis of movie reviews. *Data Science Journal*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research, 21*(140), 1–67.

Sadia, S., & Basak, R. (2021). COVID-19 sentiment classification using fine-tuned BERT. *Procedia Computer Science, 191*, 172–179.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT: A distilled version of BERT. *arXiv preprint arXiv:1910.01108*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *NeurIPS*. https://arxiv.org/abs/1706.03762

Wang, H., & Liu, M. (2024). Improved BERT-based summarization with topic embedding and contrastive loss. *Pattern Recognition Letters*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. *Proceedings of EMNLP*, 38–45. https://huggingface.co

Xie, Q., Huang, H., Saha, P., & Ananiadou, S. (2022). GRETEL: Graph-based topic-enriched summarization with BERT. *Information Fusion*, *86*, 1–12.

Xu, L., Zhang, Y., Zhang, M., & Xu, C. (2024). dimABSA: Multilingual aspect sentiment detection with BERT and LLM hybrids. *Computational Linguistics Journal*.

Xu, Y., Li, R., Zhang, L., & Liu, X. (2023). FuzzyTPBERT: Reducing redundancy in extractive summaries using fuzzy logic and topic modeling. *Information Sciences*, *613*, 165–179.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. *NeurIPS*, 5753–5763.

Zeng, F., Li, R., & Zhao, J. (2024). TP-BERT: Topic-aware and contrastive learning-based extractive summarization model. *Proceedings of ACL*.

Zhang, Y., Li, Q., & Zhou, F. (2022). DR-BERT: Dynamic reweighting for aspect-based sentiment classification. *Expert Systems with Applications*, *187*, 115849.

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2020). PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. *ICML*.

Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), e1253.

Zhou, W., & Chang, Y. (2024). Aspect-based sentiment analysis: A systematic literature review of 727 studies. *Information Processing & Management*, *61*(1), 102093.

Sommerville & Kontoya. (2019). Software engineering for machine learning: A case study. *2019 IEEE/ACM International Conference on Software Engineering (ICSE)*, 291–300.

APPENDIX

Source Code for BERT-Based Summarization and Sentiment Analysis Application

```
# app.py

import streamlit as st
from transformers import pipeline, AutoTokenizer, AutoModelForSeq2SeqLM, AutoModelForSequenceClassification
from PyPDF2 import PdfReader

from docx import Document

# Set Streamlit config (must be first)
st.set_page_config(page_title="Offline AI Summarizer", layout="wide")

# ----------------------------
# Load Offline Models
# ----------------------------
@st.cache_resource
def load_pipelines():
    summarizer_model_path = "models/led-summary"
    sentiment_model_path = "models/sentiment-model"

    tokenizer_sum = AutoTokenizer.from_pretrained(summarizer_model_path, local_files_only=True)
    model_sum = AutoModelForSeq2SeqLM.from_pretrained(summarizer_model_path, local_files_only=True)
    summarizer = pipeline("summarization", model=model_sum, tokenizer=tokenizer_sum)
```

```
    tokenizer_sent = AutoTokenizer.from_pretrained(sentiment_model_path,
local_files_only=True)
    model_sent =
AutoModelForSequenceClassification.from_pretrained(sentiment_model_path,
local_files_only=True)
    sentiment_analyzer = pipeline("sentiment-analysis", model=model_sent,
tokenizer=tokenizer_sent)

    return summarizer, sentiment_analyzer

summarizer, sentiment_analyzer = load_pipelines()

# ----------------------------
# Helper Functions
# ----------------------------
def read_pdf(file):
    pdf = PdfReader(file)
    text = ""
    for page in pdf.pages:
        text += page.extract_text() or ""
    return text

def read_docx(file):
    doc = Document(file)
    return "\n".join([para.text for para in doc.paragraphs])

def estimate_summary_length(text_length, summary_type):
    """Estimate summary length in words."""
    if summary_type == "Small":
        return min(300, int(text_length * 0.1)), 30
```

```
    elif summary_type == "Medium":
        return min(500, int(text_length * 0.2)), 80
    else:  # Large
        return min(800, int(text_length * 0.3)), 150

# ----------------------------
# Sidebar Inputs
# ----------------------------
st.sidebar.title("🗁 Document Input")

input_mode = st.sidebar.radio("Select Input Method:", ["Upload File", "Paste Text"])
summary_type = st.sidebar.radio("Choose Summary Scale:", ["Small", "Medium", "Large"])
custom_word_count = st.sidebar.number_input("Or Enter Custom Summary Word Count (optional):", min_value=50, step=10)

uploaded_file = None
text_input = ""

if input_mode == "Upload File":
    uploaded_file = st.sidebar.file_uploader("Upload a Document", type=["pdf", "docx", "txt"])
else:
    text_input = st.sidebar.text_area("Paste Your Document Text Here", height=200)

# ----------------------------
# Main App Logic
# ----------------------------
st.title("🧠 Document Summarizer & Sentiment Analyzer")
```

```
if uploaded_file:
    file_type = uploaded_file.name.split(".")[-1].lower()
    if file_type == "pdf":
        text_input = read_pdf(uploaded_file)
    elif file_type == "docx":
        text_input = read_docx(uploaded_file)
    elif file_type == "txt":
        text_input = uploaded_file.read().decode("utf-8")
    else:
        st.error("Unsupported file type.")

if text_input.strip():
    st.subheader("📄 Extracted/Provided Text")
    st.text_area("Preview:", value=text_input[:3000] + ("..." if len(text_input) > 3000 else ""), height=200)

    if st.button("🔍 Summarize and Analyze"):
        with st.spinner("Processing..."):

            doc_words = len(text_input.split())

            if custom_word_count:
                max_words = custom_word_count
                min_words = int(custom_word_count * 0.5)
            else:
                max_words, min_words = estimate_summary_length(doc_words, summary_type)

            # Approx. 1 word ≈ 1.3 tokens
            max_tokens = int(max_words * 1.3)
```

```
        min_tokens = int(min_words * 1.3)

        summary_output = summarizer(
            text_input,
            max_length=max_tokens,
            min_length=min_tokens,
            do_sample=False
        )[0]["summary_text"]

        sentiment_result = sentiment_analyzer(summary_output)[0]

    # Results
    st.subheader("📝 Summary Output")
    st.success(summary_output)

    st.subheader("📊 Sentiment Analysis")
    st.info(f"**Sentiment:** {sentiment_result['label']}\n\n**Confidence:** {round(sentiment_result['score']*100, 2)}%")
else:
    st.info("📥 Upload a document or paste text to begin.")
```