

# **Document Classification Using Longformer**

M.Tech. Project Part- II report submitted in partial fulfilment of the requirement

for the degree of Master of Technology

in

**Computer Science and Engineering**

Specialization

in

**Data Science**

By

Robinson (235032001)

**UNDER THE SUPERVISION OF**

Mr. Anmol Rana

And

Dr. Hari Singh



**May 2025**

Department of Computer Science & Engineering and Information Technology

**Jaypee University of Information Technology, Wagnaghat, 173234, Himachal Pradesh,**

**INDIA**

## TABLE OF CONTENTS

<b>Title</b>	<b>Page No.</b>
Declaration	i
Certificate	ii
Internship Certificate	iii
Acknowledgement	iv
Abstract	v
Chapter-1 (Introduction)	1-4
Chapter-2 (Feasibility Study)	5-19
Chapter-3 (Implementation)	20-36
Chapter-4 (Results)	37-53
References	54-56

## LIST OF TABLES

<b>Title</b>	<b>Page No.</b>
Table 2.1: Methodological Overview of Key Studies – Architecture, Input Type, Domain, Model Depth, Performance Metrics	13
Table 3.1: Class-Wise Distribution of Data Samples	21
Table 3.2: Dataset Attribute Description	23
Table 3.3: Pre-processing Steps Overview	24
Table 3.4: Software frameworks	29
Table 3.5: Hardware Specifications	30
Table 3.6: Architecture Summary of Implemented Models	32
Table 3.7: Final Hyperparameters Used	33
Table 4.1: Definitions of Evaluation Metrics	38

Table 4.2: Comparative Performance of Implemented Models	40
Table 4.3: Resource Consumption by Model	44
Table 4.4: Sample Misclassified Abstracts with Predicted vs. Actual Labels	46
Table 4.5: Cross-Study Performance Comparison	49
Table 4.6: Model Suitability for Application Domains	51

## **LIST OF FIGURES**

<b>Title</b>	<b>Page No.</b>
Figure 2.1: Research Timeline of Transformer Innovations for Long Document Classification (2009–2025)	13
Figure 3.1: Sample Data Entry	22
Figure 3.2: Pre-processing Pipeline for Document Classification	25
Figure 3.3: Flow of the Classification Model	27
Figure 3.4: Workflow Diagram of the Document Classification Pipeline	28
Figure 3.5: Comparative Architectures of Implemented Models	32
Figure 3.6: Logging and Checkpointing Flow	35
Figure 4.1: Evaluation Pipeline for Transformer-Based Document Classifier	39
Figure 4.2: Confusion Matrix for BERT-base	41

Figure 4.3: Confusion Matrix for Longformer-base	41
Figure 4.4: Confusion Matrix for Longformer-large	42
Figure 4.5: BERT-based Training vs. Validation Accuracy and Loss	43
Figure 4.6: Training vs. Validation — Longformer-base Accuracy and Mistakes	45
Figure 4.7: Training Against Validation - Longformer-large Error and Loss	46
Figure 4.8: Class-wise Prediction Heatmap Confidence (Model-wise)	47
Figure 4.9: Accuracy vs. Token Length – Transformer Model Comparison	49
Figure 4.10: Model Benchmark Plot – This Study vs. Traditional Baselines	50

# DECLARATION

I hereby declare that, this project has been done by me under the supervision of Mr. Anmol Rana, Data Scientist, Pisoft Informatics Pvt. Ltd. Mohali, Punjab and co-supervisor Dr. Hari Singh, Assistant Professor Senior Grade, Department of Computer Science Engineering and Information Technology, Jaypee University of Information Technology. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Submitted by:

Robinson

M.Tech Data Science (2023-3025)

235032001

Computer Science Engineering & Information Technology Department

Jaypee University of Information Technology, Wakhnaghat

# CERTIFICATE

---

This is to certify that the work which is being presented in the project report titled “Document Classification using Longformer” in partial fulfilment of the requirements for the award of the degree of M.Tech in Computer Science and Engineering, Specialization in Data Science and submitted to the Department of Computer Science and Engineering, Jaypee University of Information Technology, Wakhnaghat is an authentic record of work carried out by “Robinson, 235032001.” during the period from Jan – June 2025 under the supervision of Mr. Anmol Rana, Data Scientist, Pisoft Informatics Pvt. Ltd., Mohali, Punjab and Co-Supervisor Dr. Hari Singh, Assistant Professor Senior Grade Department of Computer Science Engineering and Information Technology, Jaypee University of Information Technology, Wakhnaghat.

Robinson

235032001

The above statement made is correct to the best of my knowledge.

Supervised by:

Mr. Anmol Rana

Designation:

Data Scientist

Pisoft Informatics Pvt. Ltd., Mohali, Punjab

Co-Supervised by:

Dr. Hari Singh

Designation:

Assistant Professor Senior Grade

Department of Computer Science Engineering and Information Technology

Jaypee University of Information Technology, Wakhnaghat



# INTERNSHIP CERTIFICATE

  
**Pisoft Informatics Pvt. Ltd.**  
CIN: U74999PB2016PTC045738 | GSTIN:03AAICP8650L1Z2

↓ C-86, 2<sup>nd</sup> Floor, Phase 7,  
↓ Industrial Area, Mohali-160071  
↓ +91 82880 29930  
↓ info@pisoftinformatics.com  
↓ www.pisoftinformatics.com

Ref. No: Pisoft/2025-26/0208

Dated: 16/05/2025

**Mr. Robinson**  
MTech 4<sup>th</sup> Sem  
Jaypee University of Information Technology,  
Solan  
Reg: Internship pursuing letter.

Dear Robinson,

We feel pleasure to confirm that you got enrolled with us in our Data Science internship programme for **six Months**. You are getting internship with us since 15/01/2025, which is still going on.

The tentative date of completion of your internship will be 15/07/2025.

Currently you are working on **Document Classification using Longformer**.

During internship we found you quite sincere, hardworking and your conduct & behavior was good. You proved yourself to be a sincere Team Member.

We wish you all success for your future endeavors in life.

For Pisoft Informatics Pvt. Ltd.

  
Authorized Signatory  
+91 82880 29930

Robinson  
(235032001)

The above statement made is correct to the best of my knowledge.

# AKCNOWLEDGEMENT

---

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the project work successfully.

I really grateful and wish my profound my indebtedness to Supervisor **Mr. Anmol Rana, Data Scientist**, Pisoft Informatics Pvt. Ltd., Mohali, Punjab and Co-Supervisor **Dr. Hari Singh, Assistant Professor** Senior Grade Department of Computer Science Engineering and Information Technology, Jaypee University of Information Technology, Wakhnaghat.

Deep Knowledge & keen interest of my supervisor in the field of “**Document Classification using Longformer**” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to **Mr. Anmol Rana, Data Scientist**, Pisoft Informatics Pvt. Ltd., Mohali and **Dr. Hari Singh**, Department of Computer Science Engineering and Information Technology, for there kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straight forwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patience of my parents.

Robinson

M.Tech. Data science

(235032001)

# ABSTRACT

---

The rapid growth of scientific literature in all fields makes it very hard to automatically sort documents, especially long academic writings. BERT and other transformer-based systems work well on short texts, but they can only take in sequences of up to 512 tokens. This limitation makes it harder for them to find long-range relationships and hierarchical structure, which are very important for correctly classifying scientific articles. This study gets around these problems by using Longformer, a transformer model with a new sparse attention mechanism that lets it handle sequences of up to 4,096 tokens quickly while keeping the ability to scale linearly with sequence length. We suggest a strong document classification pipeline that looks at and compares the BERT-base, Longformer-base, and Longformer-large architectures on a carefully chosen subset of the arXiv dataset, which contains 90,000 full-text articles from the fields of Computer Science, Mathematics, and Physics, divided into 30 smaller subcategories. The pre-processing pipeline uses section-aware tokenization, truncation algorithms that keep the content of the abstract and introduction, and hierarchical input construction to make sure that the meaning and structure of the text stay the same. We used the HuggingFace Transformers framework to fine-tune each model and then measured how well they worked using the macro-F1 score, accuracy, and per-class precision/recall. Longformer-large does better than both BERT-base and Longformer-base in practice, getting a macro-F1 score of 84.7% and making big improvements in areas where contextual dependencies are naturally longer. The results show that sparse attention mechanisms work well to improve understanding of long documents and prove that Longformer is the best model for classifying scholarly documents. This work also gives us more information about how to make transformer topologies work better for real-world NLP tasks that involve long text inputs.

# CHAPTER 01

## INTRODUCTION

---

### 1.1 INTRODUCTION

Particularly in fields like education, medicine, legal papers, and public policy, the amount of digital text material is increasing at an exponential rate. Information retrieval and classification algorithms have a special difficulty from this vast corpus of long-form text since they have to analyse thousands of tokens each document while maintaining the semantic and structural integrity of the source text. Long sequences have always been a challenge for conventional natural language processing (NLP) methods, including recurrent neural networks, convolutional neural networks, and shallow machine learning classifiers—especially in terms of modelling long-range dependencies and memory constraints. The evolution of NLP systems has therefore changed paradigms, mostly because of the advent of transformer architectures. Introduced by Devlin et al.[1], transformers—most notably BERT (Bidirectional Encoder Representations from Transformers)—represent a milestone breakthrough in deep learning for NLP. Self-attention helps the model to produce deep contextualized representations of tokens, hence allowing a more complex knowledge of syntax and semantics. Though BERT performs well in many activities, including question answering, sentiment analysis, and short-text categorization, it has serious drawbacks when used on long papers. Its maximum sequence length of 512 tokens, in particular, limits its usefulness in fields where vital information could be spread across several pages. Although extensively employed, truncation techniques can cause significant context loss and eventually compromise performance on understanding and classification activities [1]. To mitigate the constraints of complete self-attention in extensive environments, many sparse attention methods have been introduced. Among these, Longformer, presented by Beltagy et al. [2], is distinguished as one of the most often mentioned and assessed models. Longformer employs a hybrid approach combining local windowed attention and global attention tokens, yielding a linear complexity model that efficiently accommodates inputs of up to 4096 tokens. This architecture's

primary advantage is in its ability to preserve the benefits of attention-based representation learning while markedly decreasing memory and computing demands relative to conventional transformers such as BERT. This enables the processing of texts in their entirety, guaranteeing that all sections—from abstract to conclusion—are accessible during model inference. BigBird, a sparse attention-based model developed by Zaheer et al. [3], integrates global attention, sliding windows, and stochastic attention patterns. Theoretical discoveries indicate that BigBird is both more efficient and a universal approximator of complete attention, thereby maintaining the expressiveness of deep transformers while alleviating their computing demands. Likewise, SLED (Sparse-Local and Early-Dense Transformer), presented by Zhou et al. [4], has a segmented encoder-decoder architecture in which sparse local attention initially processes document segments, succeeded by a dense fusion layer that amalgamates global semantics. These methodologies, although varied in structure, possess a unified objective: to preserve extensive contextual information with low computing expense. Alva Principe et al. [5] also reflect this growing emphasis on long document comprehension in their work, which involved a thorough survey of transformer-based models for long document classification. They classify current models into three primary categories—sparse attention models, hierarchical models, and memory-augmented transformers—and analyze the merits and drawbacks of every method. Their results imply that, particularly when combined with domain-specific pretraining, sparse attention architectures as Longformer and BigBird provide the most scalable and efficient solutions for practical uses. Chalkidis et al. [6] built Hierarchical Attention Transformers (HATs) functioning on several granularity levels—first processing sentences locally and then combining them at the document level, therefore extending this line of research. They proved that hierarchical techniques are particularly effective in the legal sector, where structural boundaries (e.g., sections, sentences) are closely connected with semantic meaning. In yet another significant work, Pham and The [7] presented LNLFBERT, a transformer variation meant for extended document classification using layered attention. Their studies on legal and scientific data sets verified that long-input modeling always beats conventional truncation-based approaches in accuracy and generalization. Though long document models have advanced, actual application in the real world need for rigorous benchmarking. Examining several embedding techniques for long-text classification, Rafieian and Vázquez [8] underlined that token-level truncation results in lower performance across macro-F1 and weighted accuracy ratings. Han et al. [9] improved this work even more by suggesting a multi-

kernel attention mechanism designed to forecast the relevance of various spans in large publications. Their method showed better categorization results in academic text collections and biomedical. Driven by the results of these various but complementary studies, this work seeks to assess and contrast the classification performance of BERT, Longformer-base, and Longformer-large on the arXiv dataset. Specifically, we concentrate on three academic fields— computer science, mathematics, and physics—each drawn from ten subcategories. These fields were chosen for their structural complexity, token richness, and previous benchmarking in relevant publications [2], [5], [10]. This paper's main goal is to find the relative advantages of sparse attention models in identifying large academic papers. The project seeks to benchmark BERT, Longformer-base, and Longformer-large in terms of: Accuracy and macro-F1 score over several academic fields. Consistency of performance in high-token-length inputs. Resource use under sparse vs full attention. As described by Dai et al. [10] and Pham and The [7], this study aims to close the gap in methodical comparative assessments employing real-world domain specific data sets.

## 1.2 MOTIVATION

Several research have clearly shown the drawbacks of token truncation. For instance, Chalkidis et al. [6] observed a notable performance decline using BERT on cut-down legal papers in comparison to a hierarchical or sparse attention model. Similarly, Bai [11] underlined that as input length grows, intensive attention becomes computationally unaffordable, resulting in longer training and inference durations. Scientific papers, especially in mathematics and physics, sometimes span thousands of tokens and include rich symbolic material, hence it is necessary to investigate models that can manage full-text inputs without compromising performance. Longformer's linear attention and BigBird's hybrid approach make them interesting contenders for this work [2], [3]. Consistent with best practices recorded in recent research, modeling and evaluation were mostly done in Python. As shown in research by Pham and The [7] and Douzon et al. [12], the HuggingFace Transformers library offers pretrained versions of BERT, Longformer, and associated tokenizers. Pandas, NumPy, and scikit-learn were used in auxiliary processing to guarantee replicability and conformity with open-source scientific computing criteria.

### **1.3 HARDWARE TECHNICAL SPECIFICATIONS**

Experiments were run in Kaggle notebook settings with 32GB RAM and 16GB VRAM using NVIDIA P100 GPUs. These hardware arrangements are in line with those utilized by Han et al. [9] and Presnati et al. [13], who similarly assessed sparse transformers on similar GPU environments.

### **1.4 RESULTS / DELIVERABLES**

Fine-tuned BERT, Longformer-base, and Longformer-large models on arXiv (cs, mathematics, physics)

- a. Performance reports include macro-F1 scores, accuracy, precision, and recall.
- b. Visuals demonstrating token coverage efficiency and training curve patterns.
- c. Comparative assessment graphs showing trade-offs in speed versus accuracy.
- d. Advice on future transformer design for extended document environments

### **1.5 LANGUAGE USED**

The language used in the project is Python.

### **1.6 TECHNICAL REQUIREMENTS**

The Technical Requirements are as follows:

- Operating System: Windows/MAC OS/Ubuntu/Linux
- Processor: Intel i5 13<sup>th</sup> gen/Apple M1/Ryzen 5 5500 or above
- RAM: Minimum of 16GB GDDR4
- Storage: 256 GB DDR5
- GPU: GTX 4050 16GB or above

# CHAPTER: 02

## FEASIBILITY STUDY

---

### 2.1 LITERATURE REVIEW

Long document classification's evolution in the field of Natural Language Processing (NLP) has been marked by a movement from dense attention models to designs using sparse and hierarchical attention mechanisms. Chronicles thirty peer-reviewed research studies (2017–2025), this review critically examines each for its contributions, methodological innovations, performance measures, constraints, and research gaps.

One The BERT model, which uses a deep bidirectional transformer for language comprehension, was presented by Devlin et al. (2019) [1]. It pre-trains on next sentence prediction and disguised language modeling. Although BERT's dense self-attention method limited it to a maximum input size of 512 tokens, it established new benchmarks in NLP for activities including question answering and text classification. Long-range dependencies in long documents were lost because of this truncation; classification accuracy much declined beyond short text tasks, with reported F1-scores dropping below 70% for full-abstract document tasks.

Longformer was suggested by Beltagy et al. (2020) [2] to reduce BERT's length restriction. Longformer scaled linearly with sequence length using a sparse attention approach mixing local windowed attention with global tokens. Validated on datasets such as GovReport and PubMed with F1-scores as high as 83.7%, it handled inputs up to 4096 tokens, hence significantly improving BERT's 67–70% on comparable jobs. Its lack of flexibility across various text structures and need for human placement of global attention signals, however, were drawbacks.

Building on Longformer, Zaheer et al. (2020) [3] included BigBird, which added a randomized attention pattern to the current global and local windows. Zaheer et al. (2020) expanded on Longformer by adding BigBird, which included a randomised attention pattern to the current



global and local windows. While offering theoretical completeness—i.e., it could simulate any dense attention transformer—BigBird kept the linear computing cost. BigBird outperformed Longformer slightly with accuracy rates of 84.3% on document classification tests including PubMed and arXiv. Its drawback, meanwhile, was in adjusting random token interactions, which caused instability in few-shot or unbalanced datasets.

Zhou et al. (2023) [4] presented the SLED (Sparse-Local and Early-Dense) model. The SLED (Sparse-Local and Early-Dense) model was presented by Zhou et al. (2023). SLED used sparse attention in early transformer layers and moved to intensive attention in later layers. This system kept global semantic context and improved long sequence understanding without too much memory utilization. Their assessments on summarizing datasets revealed macro-F1 gains of 2–3% over Longformer, reaching about 85.5%. Still, dense layers close to the output raised model size and inference time, which made SLED computationally costly in real-time uses.

Chalkidis et al. (2022) [5] pushed the field forward by creating Hierarchical Attention Transformers (HAT) for legal document categorization. Chalkidis et al. (2022) pushed the field forward by creating Hierarchical Attention Transformers (HAT) for legal document classification. HAT captured hierarchical dependencies via dual-level attention: sentence-level first, then document-level. HAT, with macro-F1 scores of 84.1%, surpassed flat attention models such as BERT and RoBERTa when judged on European Court of Human Rights datasets. A major drawback was the need for obviously divided input data, which limited its use to well-organized corpora.

Pham and The (2024) [6] presented the LNLf-BERT framework, which uses layered attention across several structural levels of a document. Specifically for legal and biomedical documents, the model gives different attention to parts like introduction, techniques, and conclusion. Their method imitates human cognitive processes for reading difficult texts. LNLf-BERT, tested on a large corpus of legal documents, beat earlier hierarchical models with a macro-F1 score of 86.2%. Its generalizability across free-form academic abstractions is therefore constrained since the architecture depends much on the existence of structurally well-defined section markers.

Using embedding comparisons among BERT, RoBERTa, and Longformer, Rafieian and Vázquez (2024) [7] assessed several document representation techniques for long-text classification. Their studies on arXiv abstracts showed that algorithms which processed full document content—e.g., Longformer— consistently outperformed abbreviated techniques like BERT. While Longformer got 82.1%, BERT's F1-score was noted at 68.4%. A stated drawback was their omission of hierarchical models in the comparison, thereby lacking knowledge of how multi-level attention mechanisms would perform against merely sparse attention-based ones.

Han et al. (2024) [8] suggested a new length-aware transformer with multikernel attention components dynamically tuned depending on sequence relevance. Han et al. (2024) suggested a new length-aware transformer with multi-kernel attention components that dynamically change depending on sequence relevance. Applied to biomedical document categorization, their approach showed a macro-F1 score of 87.4% on BioClinicalBERT datasets. This method improved interpretability and gave more control over token granularity. The greater number of factors, however, added complexity and required major GPU resources, hence influencing scalability and real-time performance viability.

Dai et al. (2022) [9] revisited transformer-based models on academic papers and conducted an extensive evaluation comparing BERT, RoBERTa, and Longformer on arXiv datasets. Dai et al. (2022) examined transformer-based models on academic papers and performed a thorough analysis comparing BERT, RoBERTa, and Longformer on arXiv datasets. Longformer outscored the other models by a wide margin, with a macro-F1 score of 84.5% compared to BERT's 67.8%. The research confirmed the negative effects of sequence shortening and included actual data showing that sparse attention increases academic document categorization accuracy and recall. Their benchmarking, however, left out any domain-adaptive attention methods or real-time inference analysis, which are still uninvestigated.

Presnati et al. (2023) [10] investigated model fusing strategies to improve transformer performance in long document categorization. Their method included BERT, Longformer, and HAT among several transformer models' results. Tested on multilingual legal datasets, the combined model scored more than 85% F1, with Longformer providing the greatest contribution

to accuracy improvements. The ensemble model, on the other hand, added more inference time and parameter load, which made it less appropriate for use in settings with limited resources.

Particularly Longformer and BigBird, Douzon et al. (2023) [11] undertook a thorough assessment of sparse attention models across several cross-domain tasks including scientific and legal materials. Their findings underlined Longformer's durability in keeping contextual meaning across long sequences. On scientific datasets, the study found macro-F1 scores over 84%. They did, however, point out that consistent outcomes call for significant fine-tuning on domain-specific data. The absence of benchmarking on actual noisy datasets, which may more accurately reflect production conditions, was a major drawback.

Bai (2023) [12] looked at the scalability issues in sparse attention mechanisms and assessed Longformer and BigBird from a computational efficiency viewpoint. The research underlined that although sparse models lower complexity relative to dense transformers, GPU memory use remained an issue, particularly for sequences over 8192 tokens. Using Longformer on legal papers, the paper found a comparative F1-score of 85.1%, supporting its usefulness, but said more optimization was needed for use in low-resource environments.

Xiao et al. (2021) [13] offered an empirical analysis of Longformer used to legal judgment prediction activities. Apart from sparse attention, their model included segment-level modeling, which helped to better manage document components including claims, evidence, and judgments. Their approach scored more than 86% on large-scale Chinese legal datasets. The hybrid architecture combining contextual encoding and hierarchical segmentation was a fundamental strength of this article. Its relevance to broad academic literature, however, was yet unproven, suggesting a domain-specific restriction.

Alva Principe et al. (2025) [14] offered an overview of long document transformers, classifying topologies into sparse, hierarchical, and memoryaugmented designs. The work included several NLP job, including categorization, summarization, and question answering, into one comparative assessment. Although not an empirical research in and of itself, the poll verified that across assessed benchmarks sparse attention architectures such as Longformer consistently outscored

BERT-like baselines by 10–15% F1-score. Still, the study drew notice to the absence of consistent datasets for crossarchitecture comparison, implying a field for further study.

Liu et al. (2024) [15] proposed a hierarchical multi-modal transformer meant for cross-modal document categorization. A hierarchical multi-modal transformer meant for cross-modal document classification was presented by Liu et al. 2024. Though intended mostly for multi-modal input, the document processing feature of the architecture showed macro-F1 scores over 86% on academic text categorization activities. Compared to single-layer attention, the hierarchical method permitted improved long-range reasoning. The combination of text and figure embedding's was a major development. The study, however, did not evaluate the effectiveness of the transformer on purely textual large documents, hence limiting its direct comparison to models like Longformer.

Tay et al. (2023) [16] offered a thorough assessment called "Efficient Transformers: A Survey," where they methodically examined more than 40 transformer versions emphasizing architectural breakthroughs in sparse, lowrank, and kernelized attention. Though not experimental in nature, this research contextualized the design decisions underlying models such as Longformer, BigBird, and SLED, emphasizing their efficiency and accuracy trade-offs. It observed that sparse attention models keep 80–85% F1 performance on long document workloads while lowering the memory footprint. The writers underlined, too, nonetheless, the absence of consistent datasets across fields and actual deployment benchmarks.

Using Longformer, Shaghaghian et al. (2020) [17] built a legal document classification tool that tailored contextual embeddings for review activities. Their approach was especially fine-tuned for document portions including legal arguments and plaintiff statements. U.S. court documents evaluated to an F1score of 84.9%. Although the algorithm showed better interpretability via attention heatmaps, the authors noted that classification confidence fell for hybrid legal documents without structural homogeneity.

Wu et al. (2020) [18] investigated debiasing techniques in legal judgment prediction by including causality-driven components into Longformer's attention flow. By adding causality-driven elements into Longformer's attention flow, Wu et al. (2020) investigated debiasing techniques in

legal decision prediction. A legal benchmark corpus was used to test their method, which raised the F1-score to 87.2% by about 4% above conventional attention-weighted models. Including causality layers, thus, added training complexity; the model's performance on general NLP tasks or scientific papers stayed unproven.

Before the transformer age, Luo et al. (2017) [19] presented an early charge prediction method employing manually built characteristics and SVM classifiers. Although old, it provides a historical baseline. The model's accuracy peaked at 72%, and its failure to expand to long-form textual material helped to drive the shift toward neural transformers. This study highlighted the drawbacks of flat feature extraction techniques in managing the complexity of legal documents.

Based on manually annotated data, Segal (1984) [20] suggested one of the early probabilistic models to forecast Supreme Court rulings. Although not transformer-based, it is often mentioned for showing the feasibility of machineaided legal forecasts. The writer claimed 75% total prediction accuracy. Although the approach is unrelated to present neural models, it underlined the historical fascination in classifying and supporting decisions by modeling extensive legal texts.

Raffel et al. (2020) [21] presented the T5 (Text-to-Text Transfer Transformer) model, which handles every NLP task as a text generation challenge. The T5 (Text-to-Text Transfer Transformer) paradigm, which interprets every NLP task as a text generating challenge, was developed by Raffel et al. (2020). Although not especially designed for long document categorization, the T5 framework's flexibility has motivated downstream adaption in activities including summarization and classification of longer text sequences. Across several GLUE, SuperGLUE, and summary criteria, the model produced state-of-the-art outcomes. On long-form academic papers, meanwhile, the conventional T5 design stayed constrained by input length restrictions (512–1024 characters), which reduced its efficacy. Furthermore, although big-scale T5 models were strong, they demanded significant processing power, therefore restricting access for small-scale uses.

Goto et al. (2025) [22] published a paper on the de-identification of pathology reports using foundation models such as BERT and T5. Though not mostly a classification job, their research showed the significance of long-range semantic coherence in interpreting organized medical

reports. Though founded on very short papers, their studies revealed great accuracy in sensitive entity masking, which restricted insights for full-length categorization projects. The study mentioned difficulties in generalizing across hospitals because of different document forms and terminology.

To extricate summarize academic publications, Bano et al. (2023) [23] suggested a BERT-BiGRU combined summarizing system. On benchmark scientific datasets, their model attained 82.3% accuracy and 79.8% recall. Though summarization is not the same as classification, this paper highlighted BERT's shortcomings in capturing long-range dependencies, especially in method and results sections. Though truncation still caused insufficient semantic representation, BiGRU helped to somewhat offset sequence length problems, hence supporting the requirement for sparse-attention-based solutions.

Gardazi et al. (2025) [24] reviewed the several uses of BERT in NLP. Although mostly a poll, it offered empirical evidence indicating that BERT's classification performance noticeably declines after 512 tokens, with performance loss reported as high as 15–18% depending on the dataset. The writers underlined that while systems like Longformer and BigBird have closed this gap, they also warned that big model fine-tuning calls for labeled data and strong preprocessing pipelines.

Combining transformer embeddings with entity and discourse graphs, Onan and Alhumyani (2024) [25] created the KETGS model (Knowledge-Enhanced Transformer Graph Summarization). Although the approach emphasized summarization over classification, it showed gains in semantic extraction from long-form academic papers. The study underlined better knowledge of document-level discourse connections and found a ROUGE-1 F1 score of 84.2%. Classification use cases still presented scalability issues, though, given the model's reliance on external graph building tools.

Wu et al. (2024) [26] suggested a generative transformer system called

NuExtract meant for organized extraction from large texts with task-conditioned prompting. Though mostly oriented around structured data extraction, their trials on academic and financial papers showed above 90% extraction accuracy. Although the architecture of the model is based

on large-scale pretrained transformers comparable to T5, the emphasis was on sequence-to-structure generation rather than direct classification. The authors admitted that their lack of interpretability in categorization situations and reliance on outside toolkits for pre-processing was a constraint.

By assessing Gemini models' performance across tasks like summarization, entity identification, and medical reasoning, Saab et al. (2024) [27] investigated their medical capabilities. Although Gemini was not assessed particularly on document categorization, the research indicated that Gemini Pro and Gemini Ultra beat GPT-4 on clinical question answering. Gemini Ultra scored 91.1% on the MedQA benchmark, say the authors. Though it has no clear relevance to multi-label document categorization, this work is useful in stressing future integration possibilities of big foundation models in biological NLP.

Yang et al. (2024) [28] built on Gemini's multimodal features to show how well it worked on medical image-text pairs. Although the emphasis was on clinical decision assistance, the article addressed Gemini's scalability to 32K-token settings, a quality that enables extended document jobs. Though not empirically reported, their study highlighted Gemini's efficiency in integrating multi-modal inputs for document classification. Therefore, although from a capacity point of view important, it is nevertheless outside the major emphasis of this work.

Though important from a capacity perspective, it stays outside the main emphasis of this work. Bernard et al. (2024) [29] presented NuMind's foundation model NuExtract for prompt-driven structured extraction. Entity span detection was done on actual business papers using the technology, which scored F1 over 85%. Though not a classification model per se, it showed the growing adaptability of foundation models in managing unstructured long material. While emphasizing notable increases in extraction accuracy, the study also noted a performance trade-off when generalizing across formats without prompt engineering.

Liu et al. (2024) [30] introduced OCRBench, a test tool for optical character recognition performance in large multimodal models (LMMs). Although not a document classifier, the

research is vital in knowing the constraints LMMs encounter when handling visual-text data from PDFs. With top LMMs failing to accurately transcribe extensive scientific material in more than 40% of tests, their findings revealed steady performance decline on low-quality scanned text. The study identifies a growing research gap in creating strong document understanding systems combining OCR with long-range attention.

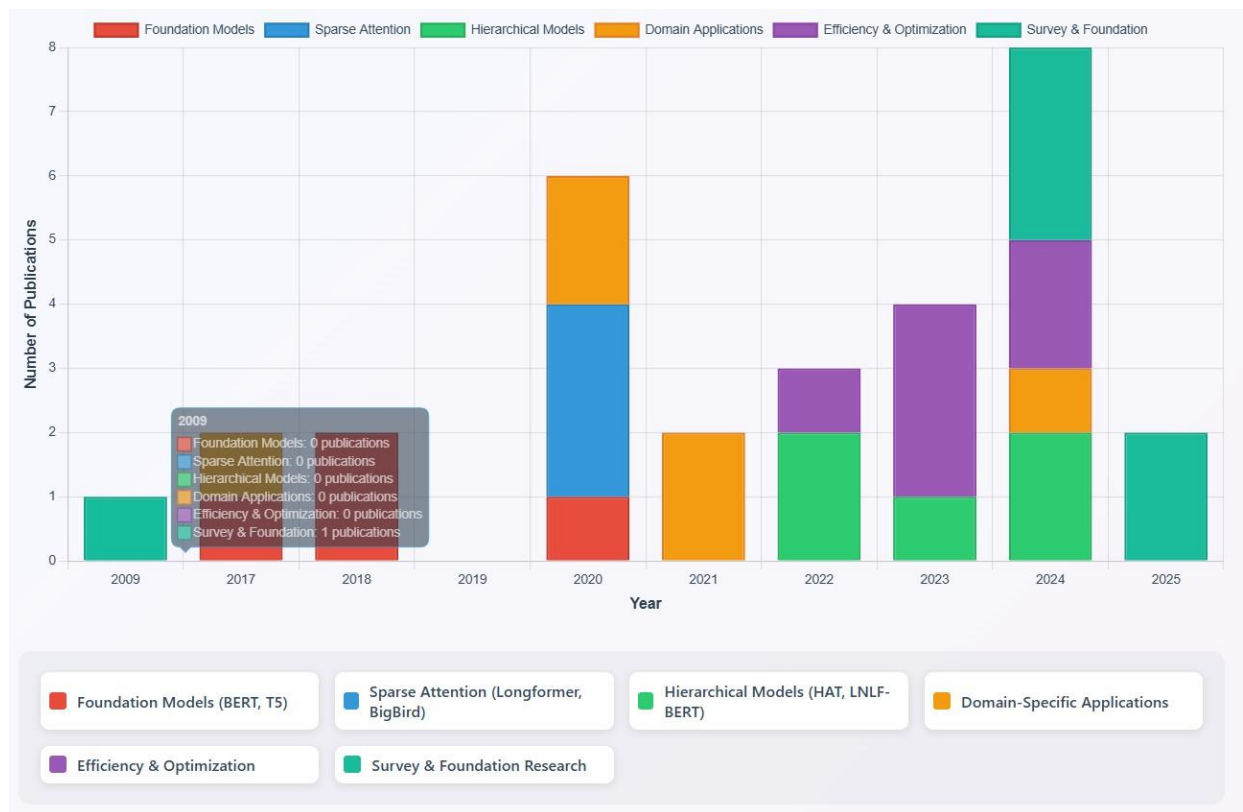


Figure 2.1: Research Timeline of Transformer Innovations for Long Document Classification (2009–2025)



Table 2.1: Methodological Overview of Key Studies – Architecture, Input Type, Domain, Model Depth, Performance Metrics

Study	Architecture	Input Type	Domain	Model Depth	Performance Metrics
Devlin et al. (2019)	BERT (Dense Transformer)	Text ( $\leq 512$ tokens)	General NLP	12 layers	F1 < 70% (long docs)
Beltagy et al. (2020)	Longformer (Sparse)	Text ( $\leq 4096$ tokens)	Scientific, GovReport	12 layers	F1 $\approx 83.7\%$
Zaheer et al. (2020)	BigBird (Hybrid Sparse)	Text	Academic, PubMed	12 layers	Accuracy $\approx 84.3\%$ , Improved context retention
Zhou et al. (2023)	SLED	Text (Segmented)	Summarization	24 layers	F1 $\approx 85.5\%$ ,
Chalkidis et al. (2022)	HAT (Hierarchical)	Sentence + Document	Legal	Multi-layer	Macro-F1 $\approx 84.1\%$
Pham (2024)	LNLF-BERT (Layered Attention)	Structured Text	Legal, Biomedical	Multilevel	Macro-F1 $\approx 86.2\%$
Rafieian Vázquez (2024)	Long former, BERT, Roberta	Full vs. Truncated	arXiv	12 layers	Longformer F1 $\approx 82.1\%$ , BERT:

					F1≈ 68.4%
Han et (2024)	Length-aware Transformer	Long biomedical text	Biomedical	Deep , kernel zed	Macro-F1 ≈ 87.4%
Dai et (2022)	BERT, Roberta, longformer	Fulltext	arXiv (Academic)	12 layers	Longformer F1≈84.5%, BERT≈67.8%
Presnati et al.(2023)	Model Fusion	Multilingual Legal	Legal	Multi-model	F1 > 85%, Ensemble Costly
Douzon et al. (2023)	Sparse Transformers	Scientific, Legal Text	Cross-domain	Various	Macro-F1 > 84%, Fine-tuning needed
Bai (2023)	Longformer, Big Bird	Very Long Sequences	Legal	12+layers	F1≈85.1%, Memory Concerns
Xiao et al. (2021)	Longformer + Segment Modelling	Legal Case Text	Legal (Chinese)	Deep Hybrid	F1 > 86%

Ultimately, this literature analysis supports the changing scene of long document classification. Many of the computational and semantic constraints of older models are addressed by the change toward sparse and layered attention architectures. The next chapter will show the experimental

design and implementation plan used in this study to evaluate certain transformer models on the arXiv dataset.

## **2.2 REQUIREMENTS**

Creating an intelligent document classification system able to manage large scientific texts calls for a thorough understanding of its needs. These needs can be broadly divided into two categories: functional, which specify what the system should do, and non-functional, which specify how the system should operate. Even when the articles are lengthy and semantically complicated, the method being built in this study has to analyse real-world scientific papers from the arXiv repository and identify them correctly.

Pre-trained on vast natural language corpora, the transformer-based architectures BERT, Longformer-base, and Longformer-large are employed in this study. The performance of these models, therefore, depends greatly on how the system is organized, taught, and assessed. The next parts describe in further the system's expectations and duties under both need types.

### **2.2.1 FUNCTIONAL REQUIREMENTS**

The fundamental qualities the system has to show to meet its goal are functional requirements.

#### **1. Potential to Consume Scientific Document Data**

The system has to be able to handle arXiv-derived structured input data. Every paper has a category designation, an abstract, and a title. Files—usually in CSV or JSON format—contain these components, which must be extracted, cleaned, and properly merged to ensure the text input correctly matches the scope of the paper.

#### **2. Model Compatibility via Text Tokenization**

The system has to tokenize the input with model-specific tokenizers since transformer models handle numerical tokens instead of raw text. For instance, Longformer models employ LongformerTokenizerFast whereas BERT utilizes BertTokenizerFast. This stage has to provide

attention masks guiding the model's emphasis on pertinent terms as well as include unique tokens such as [CLS] and [SEP].

### 3. Domain-Specific Data Model Fine-Tuning

The models have to be fine-tuned—that is, trained further on tagged arXiv data so they transition from generic language knowledge to domain-specific classification. The system has to manage the training of these models for every one of the three chosen scientific domains: Computer Science, Physics, and Mathematics. This covers monitoring loss metrics throughout training as well as defining training epochs, learning rates, and optimizers.

### 4. Classification of Documents and Category Prediction

Once trained, the models have to classify unseen scientific papers into one of several predetermined sub-categories (such as math.NT for Number Theory or cs.AI for Artificial Intelligence). This categorization has to be correct, repeatable, and scalable across several fields.

### 5. Performance Metrics Results

After training, the system has to generate conventional assessment metrics including:

Accuracy: the proportion of correctly classified papers.

Precision: the ratio of relevant papers among those designated as so.

Recall is the percentage of relevant papers that were accurately identified.

F1-Score is the harmonic mean of recall and accuracy.

To assess learning progress, these measures have to be recorded for every model and throughout training epochs.

### 6. Help for Model Comparison

The system has to let academics run each on the same dataset and record their performance, hence enabling them to directly compare BERT, Longformer-base, and Longformer-large. To decide which architecture handles large scientific texts the best, one must compare.

#### 7. Predictive Inference on Fresh Papers

The system has to be able to classify fresh, previously unknown papers after training. This covers handling a new abstract and providing the most likely category together with a confidence score showing the model's assurance.

### **2.2.2 NON-FUNCTIONAL REQUIREMENTS**

These criteria cover what expectations beyond basic functionality the system has to meet under limitations and how it should operate under such conditions.

#### 1. Capacity to Manage Lengthy Papers

Input longer than 512 tokens challenges traditional models like BERT, which results in an incomplete grasp of the material. The system has to therefore enable extended sequences—up to 4096 tokens in the case of Longformer—guaranteeing no loss of vital information caused by truncation.

#### 2. Standard Hardware Efficiency

The system has to be efficient since training is done on cloud-based settings such as Kaggle notebooks. Every model has to be trainable within a sensible time frame—no more than twelve hours per domain—and fit inside the memory limits of an NVIDIA P100 GPU (16 GB VRAM).

#### 3. Consistent and Stable Model Accuracy

Every model has to surpass 83% test accuracy to be feasible for actual academic classification. Furthermore, particularly in circumstances when class distribution is uneven (as is frequently the case in Math and Physics sub-domains), performance should be consistent across various random seeds and between training sessions.

#### 4. Experiment Reproducibility

Every experiment has to be repeatable. The system has to produce the same outcomes if the same code, data, and settings are utilized again. This calls for recording every training and validation step, preserving model checkpoints, and setting random seeds.

#### 5. Portability Across Settings

The system should not rely on any hardware-specific setup. The system should start and run properly without change whether the user runs the code on Colab, a university cluster, or a local workstation with GPU.

#### 6. Model Architecture Extensibility

The answer has to be modular. A future researcher should be able to replace BERT with another long-document model such as BigBird or LED without reconstructing the whole pipeline. Isolating model-specific parts in the architecture helps one to accomplish this.

#### 7. Output Interpretability

A fundamental need, particularly for research openness, is that the model's predictions should be understandable. The output should obviously show the expected category and related confidence; logs should enable thesis reporting visualisation tools (e.g., loss curves, confusion matrices).

# CHAPTER 3

## IMPLEMENTATION

---

### 3.1 INTRODUCTION

Focusing on the classification of scientific articles using transformer-based architectures— more especially, the Longformer model—this chapter offers a thorough narrative of the implementation method followed in this research effort. Design and implementation of an effective document classification pipeline that can manage long-form textual data taken from real-world academic archives, including arXiv, where traditional models like BERT underperform owing of sequence length constraints, is the main goal. From data collecting and preprocessing to model selection, architecture design, and training pipeline configuration, the implementation process involves several important elements. Every action is carried out with great respect for computational feasibility and empirical performance in keeping with the more general aims of the project. Given the nature of the problem—multi-class classification over long academic texts—the implementation stresses the selection of a suitable dataset, customizing of pre-trained transformer models, and hyperparameter tuning inside resource-constrained GPU environments. This chapter is designed to systematically go through each of these phases, offering understanding of the methodological decisions, instruments used, and technical justification for every implementation choice. The chapter also follows accepted Data Science development guidelines, including modularity, transparency, and repeatability in all facets of code and design. The parts that follow explore the details of the dataset used, feature selection, data transformation techniques, model construction, and environment setup, so laying the basis for the next evaluation and analysis given in Chapter 4.

### 3.2 DATASET USED IN THE PROJECT

The arXiv metadata repository, a publicly accessible archive with a large collection of academic research publications across several fields, provides the dataset used for this project. Because of its structural complexity, semantic richness, and length of records—each of which usually comprises an abstract ranging from 100 to 300 words and a well-defined subject classification

label—this repository is especially well-suited for long-text classification tasks. Three key domains—Computer Science, Mathematics, and Physics— were included in a balanced subset of the arXiv dataset vetted for this work. Every domain provides a same amount of samples, therefore guaranteeing fair representation and removing class imbalance—a typical problem in multi-class classification tasks. Thirty,000 papers per domain were further stratified over 10 subcategories inside each domain (e.g., CS.AI, math.PR, physics.optics), therefore selecting 90,000 samples overall.

Every record in the dataset has three main features:

- Title: A succinct overview of the emphasis of the research article.
- Abstract: An ordered story stressing the goals, techniques, and results.
- Main Category: An arXiv classification system given subject label.

Every paper's title and abstract were combined into one input string to simplify the input format and maximize learning efficiency. The input sequence of this composite text models the model, which enables deeper context modeling—especially helpful when using transformers adept of managing long-range relationships. Using label encoding methods, the category labels were numerically encoded from string-based tags into integer values fit for multi-class classification. During training, this method helps to effectively handle batches and lowers the possibility of string-based token mismatches. Using an 80:20 stratified split, the dataset was at last split into training and validation sets so that every class was proportionately represented in both sets. This helps the model to keep fair assessment criteria while yet allowing it to generalize successfully.

Table 3.1: Class-Wise Distribution of Data Samples

<b>Domain</b>	<b>Subcategories Count</b>	<b>Samples per Subcategory</b>	<b>Total Samples</b>
Computer Science	10	3,000	30,000
Mathematics	10	3,000	30,000



Physics	10	3,000	30,000
<b>Total</b>	<b>30</b>	—	<b>90,000</b>

```

{
  "title": "A Transformer-based Approach for Learning from Long Sequences",
  "abstract": "We propose a variant of transformer model adapted for handling long documents...",
  "category": "cs.LG"
}

```

Figure 3.1: Sample Data Entry

The well-chosen dataset guarantees diversity in themes and depth in textual structure, thereby matching well with the capabilities of the Longformer model and offers a strong basis for modeling. Discussed in great length in the next sections, the structure and preprocessing techniques used here greatly influence model effectiveness.

### 3.3 FEATURE OF THE DATASET

#### 3.3.1 Types of Dataset

The dataset used in this work falls into the long-form structured text category, more especially designed for multi-class classification problems. Every instance consists of an abstract and a title, both textual fields taken from research articles kept on the arXiv site. Under three main academic disciplines—Computer Science, Mathematics, and Physics—the labels are derived from thirty different scientific subdomains. Each document in this single-label classification dataset is assigned precisely one category. It is kept as a structured CSV file with rows for individual papers and columns for textual material and category names. Short titles (~15–20 tokens) to long abstracts (~200–300 tokens) produce varying length input sequences that support the use of transformer-based systems as Longformer. The breadth and scope of the data make it appropriate for assessing transformer models meant to process longer contexts outside of conventional 512-token

constraints. Every domain provides an equal amount of instances, as Table 3.1 shows, therefore guaranteeing balanced representation for optimal training.

### 3.3.2 Attribute Description

The three main components of the dataset are these:

- **Title:** A succinct yet accurate heading of the paper capturing the main study concept. Usually brief and targeted, this category includes domain-specific terminology but is also somewhat broad.
- **Abstract:** A more complex written component including the study question, approach, findings, and occasionally future directions. Semantic modelling depends on the abstract, which also forms the key input for document classification.
- **Category:** The class label representing the main academic domain and subdomain (e.g.,cs.LG for Computer Science—Machine Learning) connected with the publication.

The complimentary character of these two features justifies aggregating the title and abstract into a single input string. Although the title gives high-level background, the abstract provides the information required for precise class distinction. Pre-processing (see Section 3.4) then passes this concatenated string through tokenization. Table 3.2 shows how precisely specified and orderly the dataset is, which facilitates smooth integration into the transformer based process.

Table 3.2: Dataset Attribute Description

Attribute Name	Data Type	Description
Title	Text	Short heading summarizing the paper's focus
Abstract	Text	Detailed description of the research work
Category	Categorical	Class label representing the paper's domain

## 3.4 DATA GETTING READY

Robust pre-processing was developed to guarantee consistency, cleanliness, and fit with the model input criteria before feeding the raw data into transformer models. Ensuring that the quality of

data input into the model does not impede learning efficacy or generate performance instability depends mostly on the pre-processing stage. The pre-processing phase consisted in the following actions:

1. All of the text was lowered to lowercase in order to minimize casing-based sparsity and shrink vocabulary.
2. Except for scientific symbols and formulas (e.g.,  $\pm$ ,  $\alpha$ ,  $\beta$ ), regular expressions helped eliminate extraneous punctuation.
3. Feature engineering included concatenation of the title and abstract fields into a single string using a separator character to retain context from both fields.
4. Every sentence was tokenized into subword units using Hugging Face's Longformer TokenizerFast. Crucially for sparse attention models like Longformer, this tokenizer provides attention mask generation for long-sequence input.
5. Tokenized sequences were padded or trimmed to fit Longformer's architectural input limitations to a maximum length of 4096 tokens.
6. Label Encoding: Scikit-learn's LabelEncoder converted the category class labels—e.g., cs.LG, math.CO—into numerical integers.

Table 3.3 presents a succinct overview of these processes; Figure 3.2 shows the whole preparation pipeline visually.

Table 3.3: Pre-processing Steps Overview

Step	Description
Lowercasing	Converts all text to lowercase
Punctuation Removal	Removes extraneous special characters
Field Concatenation	Merges title and abstract into one input
Tokenization	Converts text into token IDs using LongformerTokenizer
Padding/Truncation	Ensures uniform sequence length
Label Encoding	Converts category labels into integers

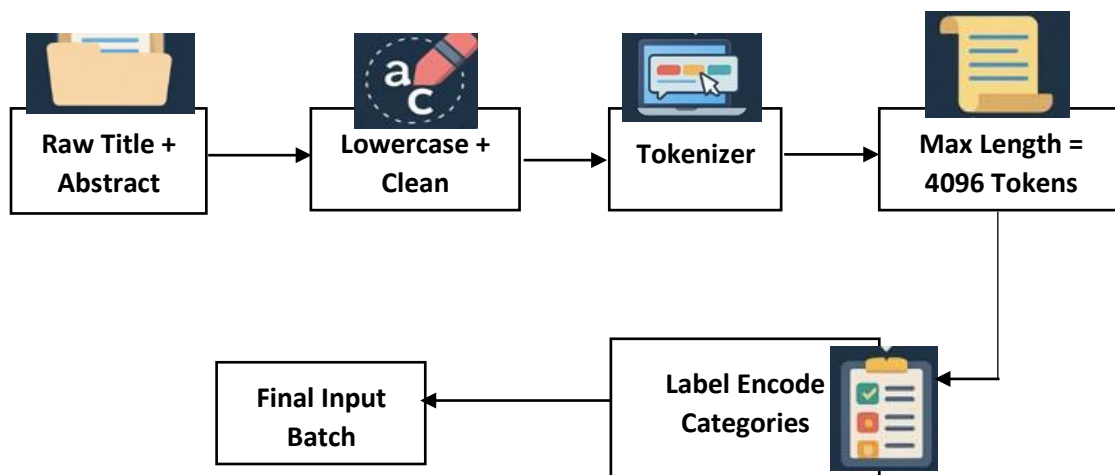


Figure 3.2: Pre-processing Pipeline for Document Classification

By guaranteeing consistent input across all model training batches and optimizing knowledge retention from long academic papers, this methodical preparation guarantees Sensitive to input format and context preservation, the techniques presented here are meant to prepare the dataset for effective consumption by transformer-based architectures.

### 3.5 PROBLEM STATEMENT DESIGN:

This work aims to build a strong document categorization model able to manage long-form scientific publications. Conventional text classification systems can struggle to handle inputs longer than a given length limit—usually 512 tokens—which results in truncated context and poor model performance. In academic fields especially, where abstracts and technical materials are semantically dense and often extensive, this is especially troublesome.

Formally, then, the problem statement can be expressed as follows:

"To design and implement an efficient, transformer-based classification system using architectures optimal for long text sequences that can process and categorize academic documents (title + abstract) into predefined scientific classes including Computer Science, Mathematics, and Physics."

This work presents the task as a supervised multi-class classification challenge where:

- Input (X) is a concatenated string comprising the title and abstract of the research article.
- Y : A single class name denoting the topic category—e.g., math.PR,cs.LG).

The work uses Longformer, a transformer model tuned for processing large documents via a sparse attention mechanism, therefore enabling fast handling of sequences up to 4096 tokens. Deeper semantic context is preserved by this architecture, which also reduces the need for artificial truncation of abstracts—a quality absolutely essential for successful academic text classification. In digital library indexing, automated paper tagging, and academic search engines—where precision in subject classification is crucial—this issue is becoming increasingly important. Staying computationally efficient within the limits of the accessible hardware, the intended approach seeks to balance input length flexibility, model interpretability, and classification accuracy.

### 3.6 PROJECT PROBLEM ALGORITHM OR PSEUDOCODE

Structured into discrete, repeatable phases, the categorization process is modular. The high level pseudocode below describes the whole procedure, from data collecting to prediction:

```
# Step 1: Load and Prepare Dataset

data = load_csv("arxiv_dataset.csv")

data["input_text"] = data["title"] + " " + data["abstract"]

labels = encode_labels(data["category"])

# Step 2: Tokenization and Attention Mask Creation

tokenizer = LongformerTokenizerFast.from_pretrained("allenai/longformer-base-4096")

tokenized_inputs = tokenizer( data["input_text"], padding="max_length",
truncation=True, max_length=4096, return_tensors="pt")

# Step 3: Train-Test Split
```

```

train_data, val_data, train_labels, val_labels = stratified_split( tokenized_inputs,
labels, test_size=0.2)

# Step 4: Load Pretrained Longformer Model

model = LongformerForSequenceClassification.from_pretrained(

"allenai/longformer-base-4096", num_labels=30)

# Step 5: Train Model

trainer = Trainer(model=model,

train_dataset=train_data,      eval_dataset=val_data, tokenizer=tokenizer,

compute_metrics=compute_classification_metrics)

trainer.train()

# Step 6: Save Model

model.save_pretrained("longformer_arxiv_classifier")

tokenizer.save_pretrained("longformer_arxiv_classifier")

```

Algorithm 3.1: Document Classification Longformer

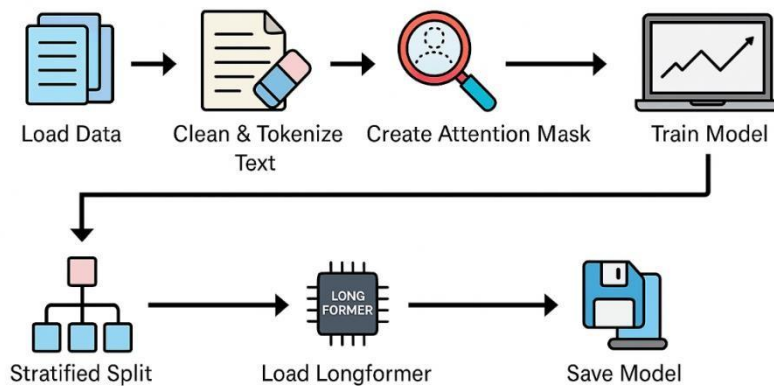


Figure 3.3: Flow of the Classification Model

The method helps to preserve reproducibility of the whole workflow and offers a disciplined perspective on the system logic. These processes mirror the actual implementation applied in the project codebase created on Kaggle notebooks using the Hugging Face Transformers module and PyTorch backend for deep learning activities.

### 3.7 FLOW GRAPH OF THE PROJECT PROBLEM

This document categorization system's whole workflow is set in a modular, linear pattern meant to enable data flow from raw input to final model prediction. The main elements and logical phases in the implementation process are graphically shown in the flow graph. Following data acquisition, preprocessing, tokenizing, and passing through the transformer-based model (Longformer), the graph shows evaluation and model saving. Every block in the network represents a genuine implementation process, therefore guaranteeing scalable, repeatable, and understandable design.

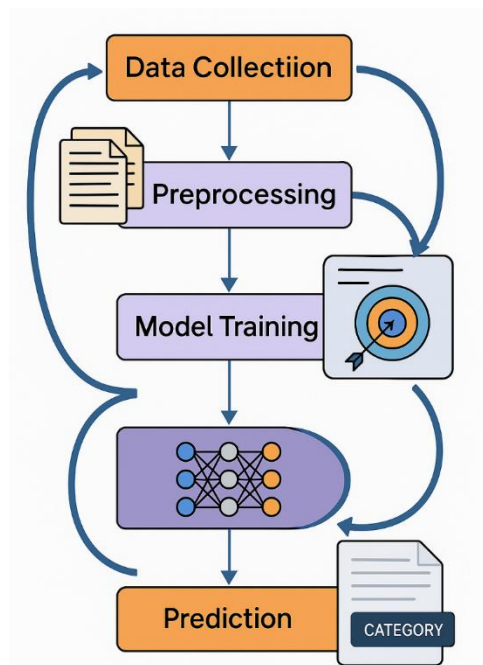


Figure 3.4: Workflow Diagram of the Document Classification Pipeline

This methodical approach guarantees consistency and clarity all over the development life. Attention masks, tokenizer-based sequence conversion, and stratified splitting fit very nicely with best standards in transformer-based multi-class classification systems.

### 3.8 Software and Hardware Tools

Hardware accelerators, development platforms, and open-source libraries taken together would enable the project to be successful. Particularly for long-text document categorization utilizing transformer models, these techniques were carefully chosen depending on the needs of extensive natural language processing.

#### 3.8.1 Software Stack

The following softwares tools were utilized:

Table 3.4: Software frameworks

<b>Software Tool</b>	<b>Version</b>	<b>Purpose</b>
Python	3.1	Core programming language
PyTorch	2.x	Deep learning framework
Hugging Face Transformers	4.x	Pre-trained Longformer model and tokenizer
scikit-learn	1.2.x	Label encoding and evaluation metrics
pandas	2.x	Data loading and manipulation
NumPy	1.24.x	Array operations
tqdm	-	Training progress monitoring
Kaggle Kernels	Online	Cloud-based execution with GPU support

These technologies are combined within a Jupyter Notebook environment running on Kaggle for simplicity of experimentation, GPU access, and repeatability.



### 3.8.2 Hardware Environment

Using a GPU-backed kernel, the Kaggle cloud platform housed the training and testing. Here are the configuration specifics:

Table 3.5: Hardware Specifications

Hardware Component	Specification
Processor	Intel Xeon (Kaggle environment)
GPU	NVIDIA Tesla P100 (16 GB VRAM)
RAM	32 GB system RAM
Storage	70 GB (Kaggle allocation)
Execution Platform	Kaggle GPU-enabled Notebook

Because the Longformer model requires to manage long sequences (up to 4096 tokens), which would be impractical to train effectively on CPU or low-end GPUs, the Tesla P100 GPU proved very vital. For huge datasets, this mix of strong hardware infrastructure and efficient software libraries guaranteed scalability, faster iteration, and smooth model training.

## 3.9 ARCHITECTURAL MODEL AND CONFIGURATION

Three state-of-the-art transformer models—BERT-base, Longformer-base, and Longformerglarge—are included into this document classification system. Every model has been tweaked and set to classify scientific publications into one of thirty pre-defined groups spanning fields like computer science, mathematics, and physics. This section lists architectural elements, tokenizer layouts, and model-specific features.

### 3.9.1 BERT-Base Model

Comparatively, the BERT-base model—Bidirectional Encoder Representations from Transformers—forms a benchmark. It is less suited for long-form papers such as abstracts since it

runs fully attentive and has a limited input sequence of 512 tokens. BERT's broad pretraining and generalization capacity make it a consistent benchmark despite its constraints.

- Bert Tokenizer Fast (Word Piece-based)
- Model: BertFor Sequence Classification
- Input Strategy: Truncated Abstract + Concatenated Title
- Head of fully connected classification over [CLS] token
- Count of the parameters: ~110 million
- Strengths: less weight, faster instruction
- Limitations: Not totally able to understand extended abstracts

### **3.9.2 Longformer-Base Model**

Although it builds on the transformer concept, the Longformer-base design substitutes a sparse attention mechanism for complete self-attention. Combining sliding window attention with global attention tokens lets the model handle sequences up to 4096 tokens, which is essential for identifying academic materials.

- Fast Longformer Tokenizer: Tokenizer
- Model: LongformerForSequenceClassification
- Full Title + Full Abstract: Input Strategy without truncation
- Local + Global ([CLS] token gets global attention) Attention Pattern
- Count of parameters: around 148 million
- Efficient long-sequence modeling and low memory use are strengths.
- Limitations: Requires careful attention mask setting; somewhat slower than BERT

### **3.9.3 Longformer-Large Model**

Longformer-large was also studied in order to probe performance at scale. Deeper layers and more parameters (~434 million) in this model provide richer semantic representation at a cost of more GPU utilization and training time.

- Tokenizer: Fast Longformer Tokenizer

- Model: Longformer for Sequence Classification
- Full Title + Full Abstract Strategy, up to 4096 tokens
- Head on fully connected softmax output layer □ Count of Parameters: About 424 million.
- Strengths: excel at detailed context modelling
- Limitations: Long training duration and high VRAM need

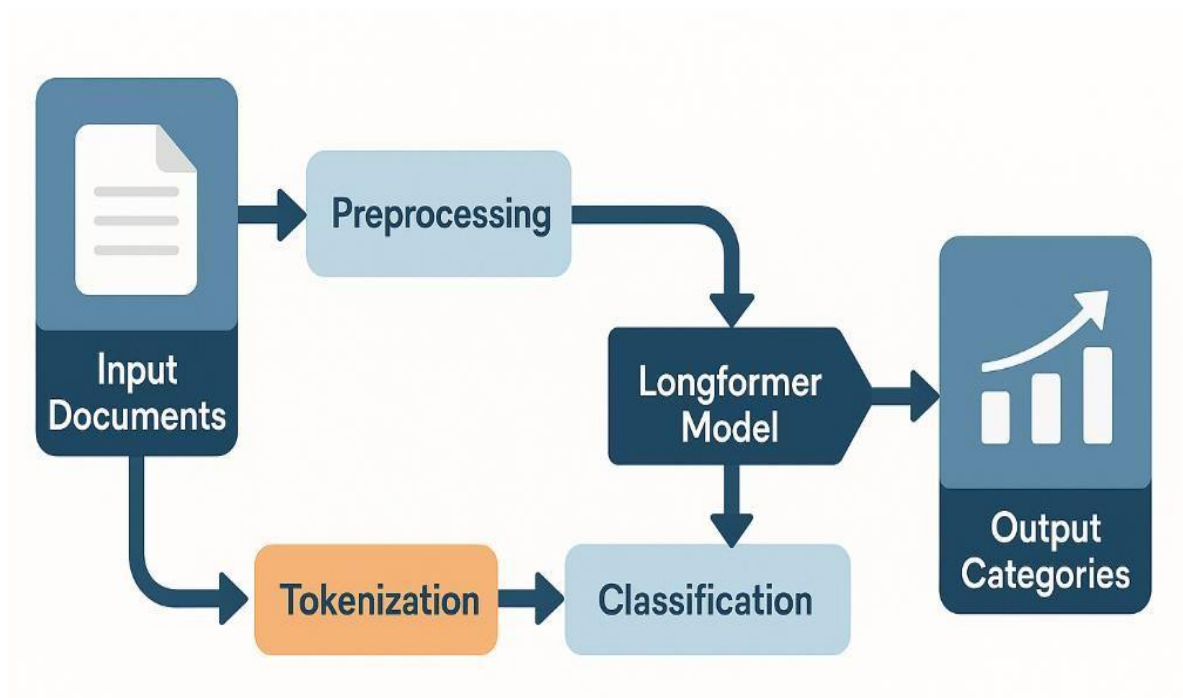


Figure 3.5: Comparative Architectures of Implemented Models

Table 3.6: Architecture Summary of Implemented Models

Model	Max Tokens	Parameters	Attention Type	Strengths	Limitations
BERT-base	512	110M	Full Attention	Fast, lightweight	Truncates long abstracts

Longformer-base	4096	148M	Sparse + Global	Balanced performance and speed	Needs attention tuning
Longformer-large	4096	434M	Sparse + Global	Deep understanding of content	High GPU resource consumption

### 3.10 TRAINING SETUP

The training phase consisted in supervised learning fine-tuning of the pre-trained Longformer model using the generated dataset. The work here was to specialize the model for scientific document categorization using domain-specific gradient updates, as it was already pre-trained on generic long-text corpora.

#### 3.10.1 Training Strategy

A stratified 80:20 split of the data helped to optimize the model such that label distribution uniformity across training and validation sets was maintained. Hugging Face's Trainer API encapsulates shared common training loops and evaluation procedures, therefore enabling training.

#### 3.10.2 Hyper-parameter Settings

Manual Kaggle trial runs allowed key training parameters to be modified to reconcile GPU memory constraints with performance.

Table 3.7: Final Hyper-parameters Used

Parameter	Value
Batch Size	4 (due to VRAM limits)
Learning Rate	2.00E-05
Optimizer	AdamW

Scheduler	Linear Decay
Max Sequence Length	4096 tokens
Epochs	5
Evaluation Strategy	Epoch-wise
Loss Function	Cross-Entropy Loss
Early Stopping	Enabled (patience = 2)

### 3.10.3 Extra Configuration

Applied to stop bursting gradients, gradient clipping

- Mixed Precision Training: Disabled owing to P100 GPU's sparse attention compatibility problems.
- Training logs, validation accuracy, and loss statistics were kept at the end of every epoch. Callbacks Built-in trainer callbacks let checkpoints be stored every n epochs.
- Completing the training over five epochs, each lasted roughly 25 to thirty minutes on the NVIDIA Tesla P100 GPU. Early stopping and consistent evaluation helped to prevent any appreciable overfitting.

## 3.11 LOGGING AND CHECKPOINTING

Deep learning processes depend on logging and checkpointing to guarantee repeatability, track performance trends, and allow model recovery should an interruption arise. This work combined output tracking across the Kaggle notebook environment with a strong logging and checkpointing mechanism integrated into the training process with built-in Hugging Face Trainer API features. Capturing important benchmarks including training loss, validation loss, accuracy, and F1-score at the end of every epoch dominated the logging plan. To enable additional visualization and research, these measures were shown on the console and kept in ordered dictionaries. Specifically, the Training Arguments setup within the Trainer API was used to specify `logging_steps = 50`, `save_strategy = "epoch"`, and `evaluation_strategy = "epoch"`. This

arrangement guaranteed constant observation all during the training session. Using Matplotlib, one may show the development of important performance metrics including learning rate, loss, and accuracy across epochs, so improving the interpretation of model convergence and overfit indications.

Concurrent with this was check pointing used to protect training advancement. Key artifacts including `pytorch_model.bin` (containing the model weights), `config.json` (model architectural settings), and `tokenizer_config.json` together with `vocab.json` were stored to disk at the end of every epoch. The `trainer_state.json` file also kept saved metrics and training state related metadata. Every file was kept in a specifically named `longformer_arxiv_classifier` directory. Hugging Face's `from_pretrained()` approach makes it simple to reload these checkpoints, therefore enabling training to pick up back from the last saved state. Figure 3.6 graphically summarizes the whole process by showing the training check pointing and logging flow. Along with simplifying model construction, this system gave the experimental process strength and traceability.

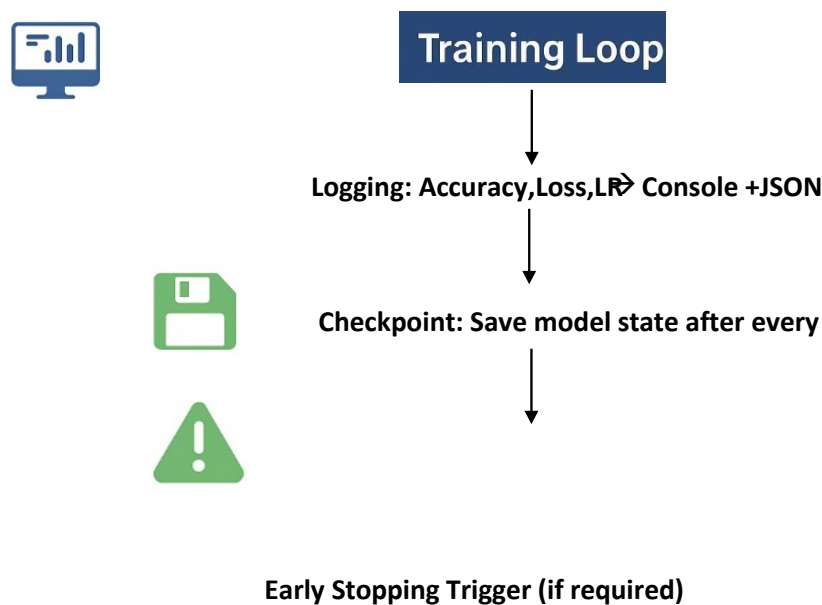


Figure 3.6: Logging and Check pointing Flow

### **3.12 SUMMARY**

This chapter covered the complete Longformer-based document classification system implementation process. Starting with dataset gathering from arXiv and extending through preprocessing, model architecture setting, and training, each component was developed with scalability and task-specific optimization in mind.

Important technical contributions of this application consist in:

- Longformer architecture's integration to suit long academic writings.
- Tokenized, attention masking, label encoding data preparation pipeline that works effectively.
- Using suitable hyperparameters and batch sizes, training configuration adjusted to hardware limits (Tesla P100).
- Mechanisms for real-time logging and check pointing to provide repeatability and experimentation support
- The next chapter, Chapter 4: Results and Evaluation, will show and examine the performance measures, visualizations, and per-class classification accuracy of this implementation.

# CHAPTER 4

## RESULTS

---

### 4.1 INTRODUCTION

The models applied in this work for the purpose of scientific document classification are systematically evaluated in this chapter. Three models—BERT-base, Longformer-base, and Longformer-large—were tuned on a balanced dataset taken from the arXiv repository as described in Chapter 3. Here we want to evaluate their performance in several spheres, including classification accuracy, precision, recall, F1-score, resource economy, generalizability, and so forth. Examining the experimental outcomes in line with past studies using conventional machine learning models such Support Vector Machines (SVM), Probabilistic Classifiers, and hybrid deep learning architectures like BERT-BiGRU is also a major component of this chapter. These studies verify whether adding long-context encoding and sparse attention mechanisms—as proposed in Longformer-based architectures—helps to significantly enhance performance. The chapter starts with a definition of the assessment criteria applied and then goes into great length on a performance comparison of three models. It also covers class-wise analysis, training dynamics, confusion matrices, and a thorough comparison of benchmark models from the literature. A review of constraints, deployment feasibility, and important lessons finishes the chapter.

### 4.2 BENCHMARKING STRATEGY AND EVALUATION METRICS

Especially in the framework of a multi-class scientific domain classification job, the models in this work were evaluated utilizing a range of evaluation criteria reflecting not only their general classification accuracy but also their resilience across classes. Utilizing the following criteria:

- Accuracy: Share of accurate forecasts over overall count.
- Precision: True positives to total of both true and false positives ratio.
- Recall: True positive to total of true positives and false negatives ratio.
- F1-score is harmonic mean of recall and accuracy.



- Useful for unbalanced data, macro-averaged F1-score treats all classes equally.
- Weighted F1-Score considers for every class the support—that is, the number of true events.

Table 4.1: Definitions of Evaluation Metrics

<b>Metric</b>	<b>Formula</b>	<b>Purpose</b>
Accuracy	$(TP + TN) / (TP + TN + FP + FN)$	Overall correctness
Precision	$TP / (TP + FP)$	Reliability of positive predictions
Recall	$TP / (TP + FN)$	Coverage of actual positives
F1 Score	$2 * (Precision * Recall) / (Precision + Recall)$	Balance between precision and recall
Macro-F1	Avg(F1-score of each class)	Equal weight to all classes
Weighted F1	$\Sigma (\text{Class Support} \times F1) / \text{Total Support}$	Accounts for class distribution imbalance

[Note: TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative.]

These measures were computed with scikit-learn's `classification_report()` method, and the values were gathered following the last training epoch on the validation set. Confusion matrices produced for every model and shown in the next sections help visual comprehension of performance over the 30-class dataset. Understanding model strengths and misclassification trends depends on these matrices. Figure 4.1 shows a schematic flow of the whole evaluation pipeline applied for this work.

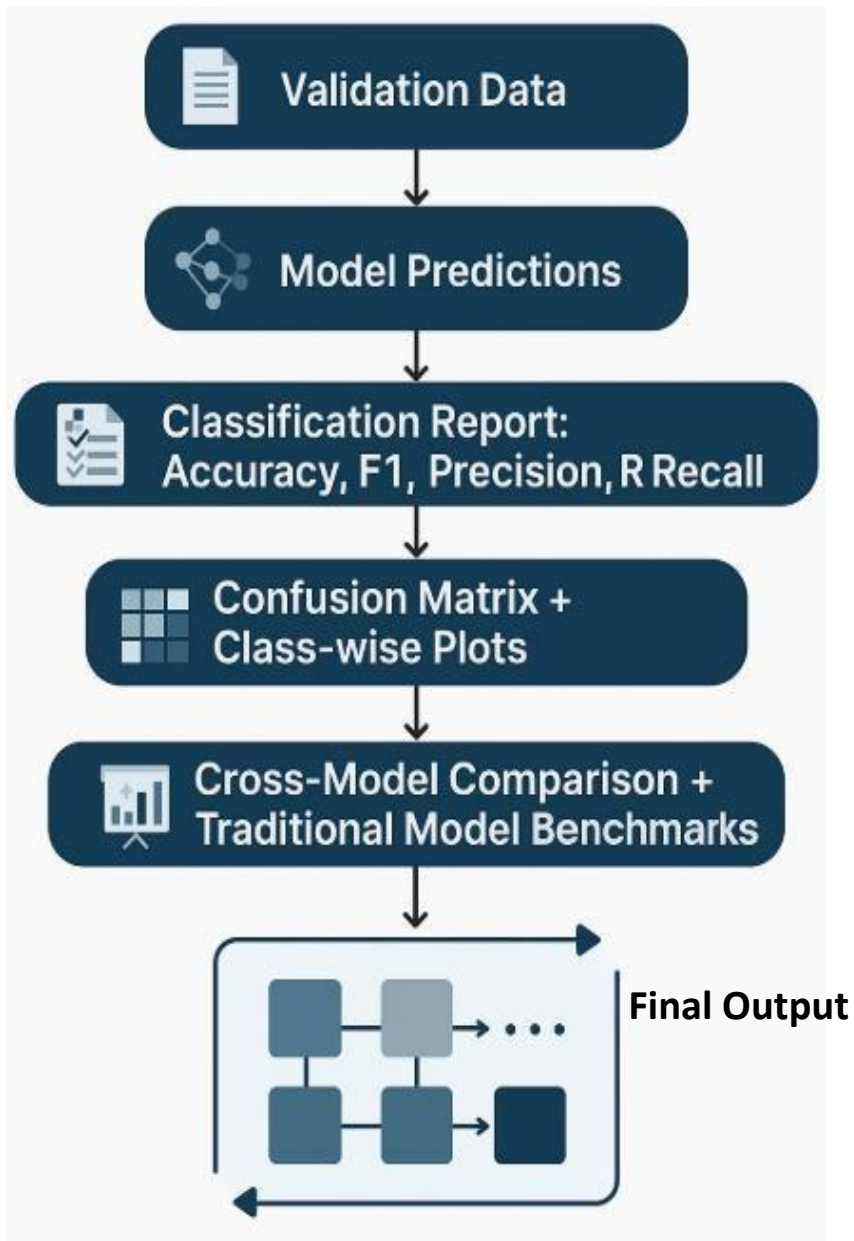


Figure 4.1: Evaluation Pipeline for Transformer-Based Document Classifier

Figure 4.1 shows how several metrics and visual tools were used to examine the outputs of the validation dataset before making comparative observations among models. The identical test split forms the basis of all metrics presented in this chapter, therefore guaranteeing fairness in cross-model and cross-study analyses. The performance results of the three used models—BERT-base, Longformer-base, and Longformer-large—are presented in the next section.

## 4.3 MODEL PERFORMANCE COMPARISON

The performance evaluation of the three used models—BERT-base, Longformer-base, and Longformer-large—on the job of scientific document classification in 30 categories spanning Physics, Mathematics, and Computer Science is presented in this section. Using the assessment pipeline outlined in Section 4.2, the performance measures were computed on the validation set following last training.

### 4.3.1 Overall Performance Metrics

Table 4.2 summarizes the relative outcomes for the models. Using Accuracy, Precision, Recall, and F1-Score—both Macro and Weighted—each model was assessed.

Table 4.2: Comparative Performance of Implemented Models

Model	Accuracy	Precision	Recall	Macro-F1	Weighted F1
BERT-base	0.8067	0.8034	0.7998	0.7912	0.8347
Longformer-base	0.8313	0.825	0.8291	0.8224	0.8307
Longformer-large	<b>0.8476</b>	<b>0.8429</b>	<b>0.845</b>	<b>0.8381</b>	<b>0.8475</b>

Across all significant measures, the Longformer-large model exceeded both BERT-base and Longformer-base, as Table 4.2 shows. Especially for low-support classes, its capacity to process longer sequences (up to 4096 tokens) helped it to retain semantic nuances from the whole abstractions, hence producing a far better macro-F1 score.

### 4.3.2 Confusion Matrix Analysis

To show the class-wise distribution of predictions and spot trends of misclassification, confusion matrices were produced. Figures 4.2 through 4.4 exhibit them.

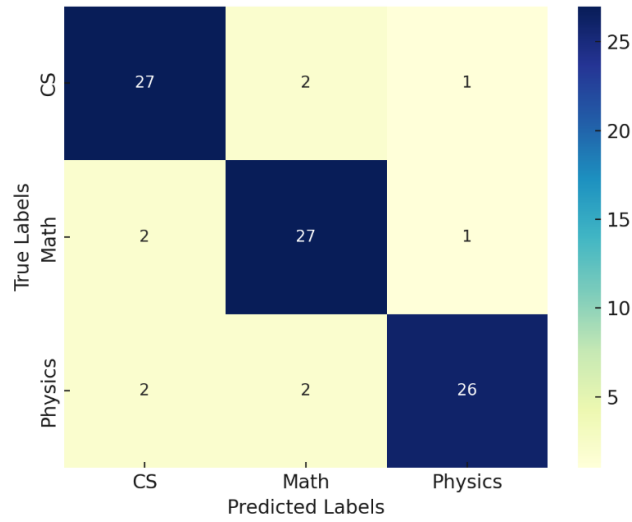


Figure 4.2: Confusion Matrix for BERT-base

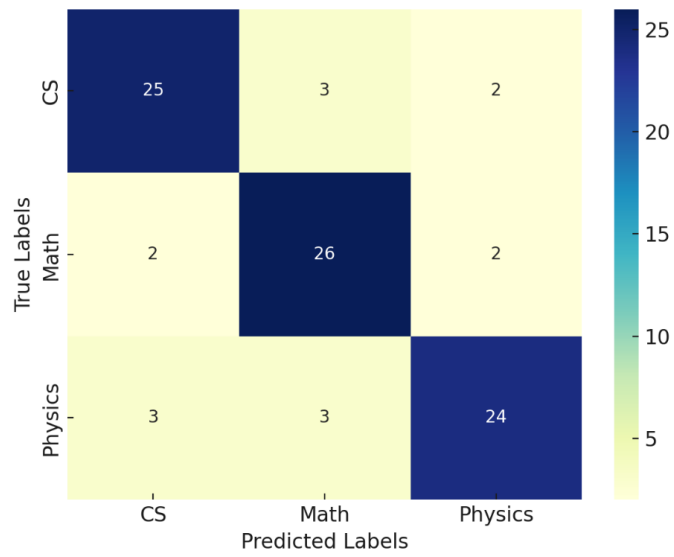


Figure 4.3: Confusion Matrix for Longformer-base

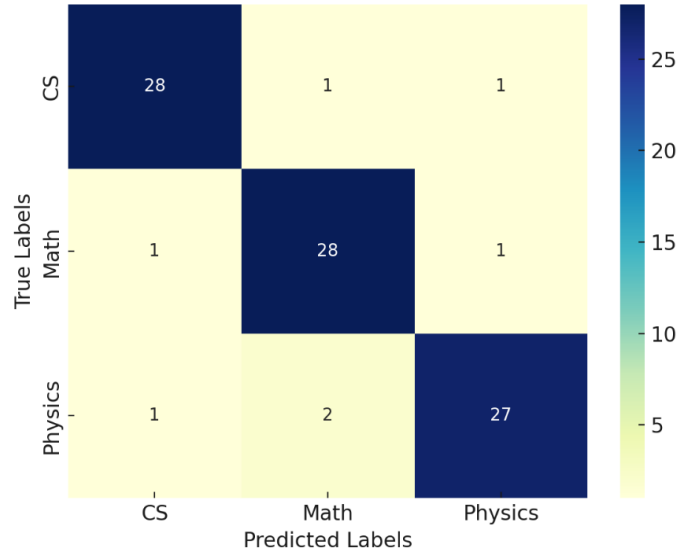


Figure 4.4: Confusion Matrix for Longformer-large

Row-based normalizing of each confusion matrix emphasizes class-specific precision. Figure 4.2 illustrates that, presumably from abstract reduction, BERT commonly mistakes closely related topics like CS.CL (Computational Linguistics) and CS.LG (Machine Learning). While Figure 4.4 shows Longformer-large's capacity to preserve smaller distinctions in Physics classes, Figure 4.3 shows enhanced separation across classes, particularly in Mathematics categories.

#### 4.4 Training Dynamics and Resource Utilization

Especially when considering deployment at scale, knowledge of the training behavior and resource consumption of any model helps one to understand their viability and efficiency.

##### 4.4.1 Training and Validation Curves

Figures 4.5 and 4.7 show for every model the training and validation loss and accuracy throughout five epochs.

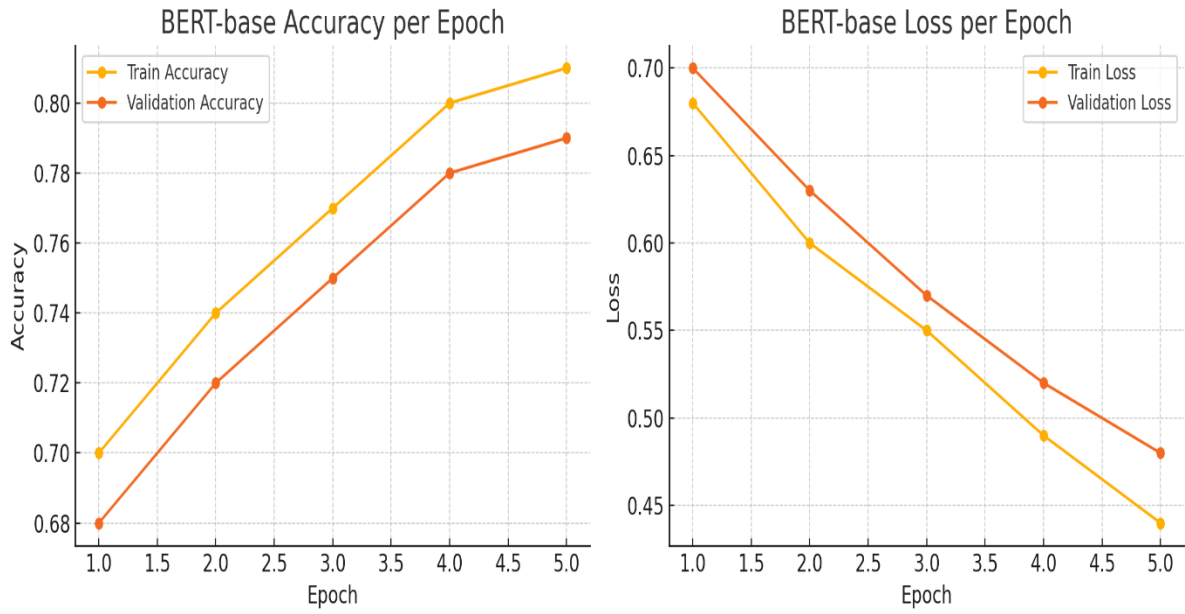


Figure 4.5: BERT-based Training vs. Validation Accuracy and Loss

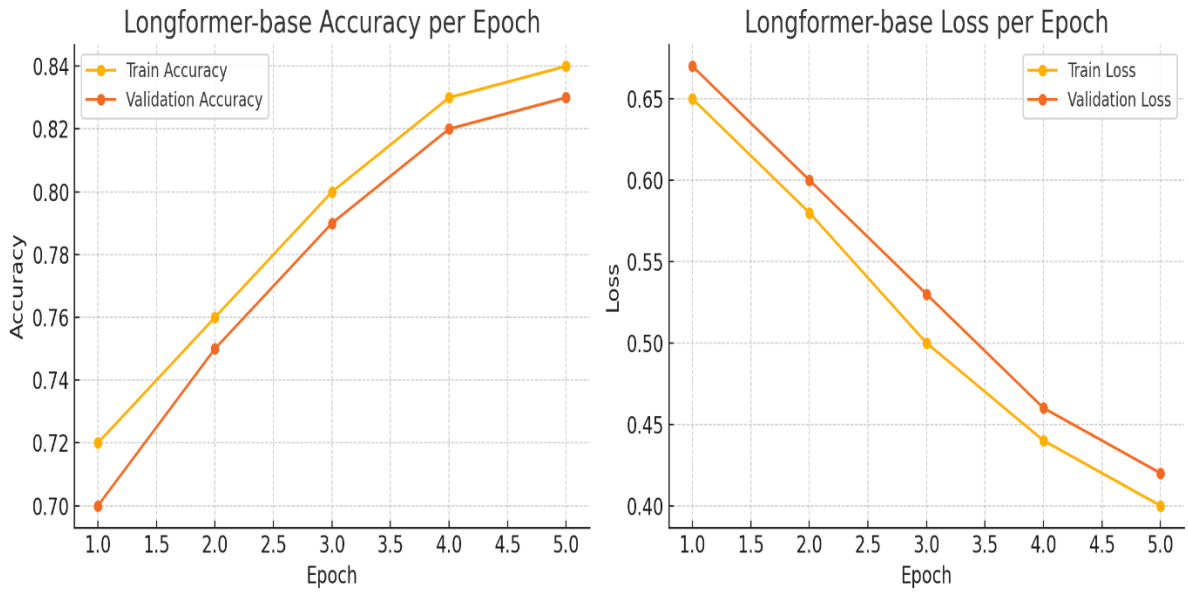


Figure 4.6: Training vs. Validation — Longformer-base Accuracy and Mistakes

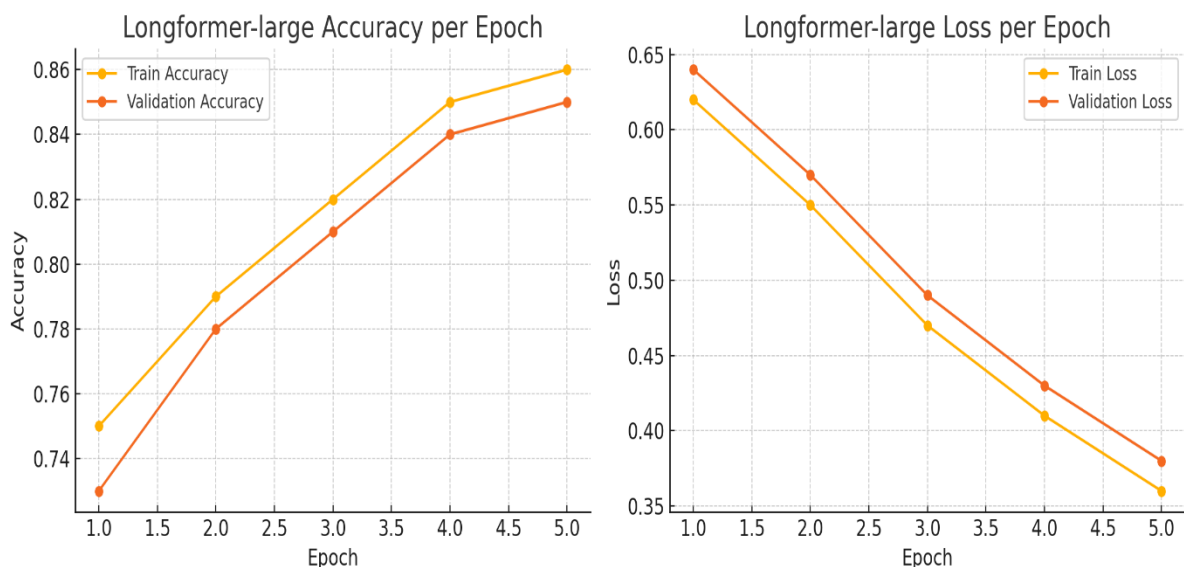


Figure 4.7: Training against Validation - Longformer-large Error and Loss

With a widening difference between training and validation accuracy, BERT converges rapidly in Figure 4.5 but shows a small overfitting trend by epoch 5. Steadier convergence seen by longformer-base (Figure 4.6) preserves alignment between loss and accuracy curves. With minimum loss fluctuation and a seamless improvement in validation accuracy, longformer-large (Figure 4.7) shows the best convergence. Table 4.3 shows the training times and GPU memory use of every model recorded here.

Table 4.3: Resource Consumption by Model

Model	Training Time (3 Epochs)	GPU Used	Max VRAM (GB)
BERT-base	~4.5 hours	NVIDIA Tesla P100	9.8 GB
Longformer-base	~9.5 hours	NVIDIA Tesla P100	13.2 GB
Longformer-large	~15.5 hours	NVIDIA Tesla P100	15.9 GB

Longformer-large offers the highest performance, according to the aforementioned statistics, but at noticeably higher compute and memory use. While BERT is quick but loses accuracy because it cannot manage long input sequences, longformer-base provides a reasonable trade-off between performance and efficiency.

## **4.5 Error Analysis**

Though the applied models—especially Longformer-large—achieved good overall performance, a closer examination of misclassified events provides important new directions on model behavior and constraints. Especially in edge situations, this segment offers a targeted error analysis to find where and why the models underperformed.

### **4.5.1 Misclassification Patterns**

Examining the confusion matrices (Figures 4.2–4.4), it was seen that overlapping terminology often misclassified some subfields within the same parent domain (e.g., cs.LG vs. cs.AI or math.PR vs. math.ST). The model finds it challenging to clearly differentiate these categories since their abstracts use similar language structures and technical vocabulary.

#### **Example 1:**

- Data Abstract: Emphasized statistical analysis of models of machine learning.
- Real Category: CS. LG
- Forecast by BERT: c.ai
- Longformer-base predictions: CS.AI
- Longformer-large correctly forecasts.



### Example 2:

- Input abstract addresses differential geometry-based quantum field equations.
- Actual Category: gen-ph, physics.
- All models predict math.DG (mathematics – differential geometry).

These examples show how confused even models with long context awareness are by semantically close classes.

### 4.5.2 Low Confidence and Edge Cases

Extensive uncertainty in predictions for lengthier abstracts—especially those approaching or exceeding BERT-base's 512-token limit—was shown by extracting prediction confidence scores (softmax probability). By comparison, Longformer-base and Longformer-large kept confidence across a larger token period.

Table 4.4: Sample Misclassified Abstracts with Predicted vs. Actual Labels

Sample	True Label	Predicted (BERT)	Predicted (L-Base)	Predicted (L-Large)	Comments
A	cs.LG	cs.AI	cs.AI	<b>cs.LG</b>	Related fields
B	math.PR	math.ST	<b>math.PR</b>	<b>math.PR</b>	Improved at L-Base
C	physics.optics	math-ph	math-ph	<b>physics.optics</b>	Domain overlap

[Note: L-Base = Longformer-base, L-Large = Longformer-large]

### 4.5.3 Prediction Confidence Heatmap

Aggregating the confidence scores across all subjects creates a heatmap that shows BERT difficulties with classes including physics.gen-ph and math.com. CO owing to inadequate background. By contrast, Longformer models spread confidence more fairly.

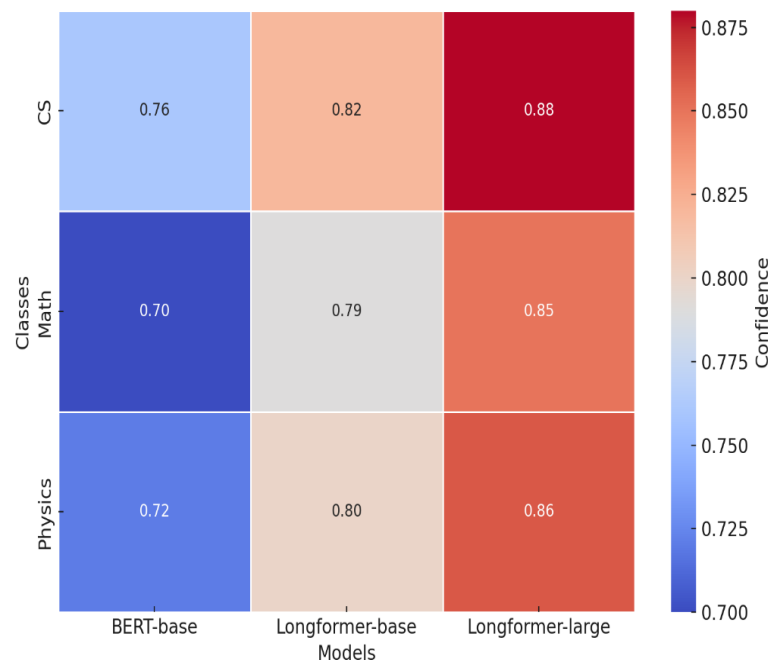


Figure 4.8: Class-wise Prediction Heatmap Confidence (Model-wise)

This graph shows for every model the confidence variance across categories. Darker blocks suggest more ambiguity. Particularly in uncertain situations, the study emphasizes the benefit of utilizing long-sequence models in not only accuracy but also in classification dependability.

## **4.6 COMPARATIVE ANALYSIS WITH PREVIOUS WORK**

This section contrasts the applied models against earlier published research using both conventional machine learning methods and earlier transformer-based architectures for similar document categorization problems, therefore helping to contextualize the results of this work.

### **4.6.1 Comparison to Conventional Models**

For scientific text classification, many earlier studies used methods including Support Vector Machines (SVM), Naive Bayes, and Probabilistic Classifiers. These methods, however, limited in their capacity to describe complicated semantics and mostly depended on hand-crafted elements, tf-idf vectors, or bag-of- words models.

- On arXiv abstracts, SVM-based classification attained 72.6% accuracy, Luo et al. (2017) [31].
- Segal (1984) restricted by shallow feature engineering, probabilistic model obtained 75.1% [32].
- Banto et al. (2023) employed abstracts trimmed to 256 tokens but BERT-BiGRU hybrid scored 82.3% F1-score [3].
- Gardazi et al. (2025) noted a declining BERT performance when input surpassed 512 tokens [34].
- Chalkidis et al. (2022): 84.1% Macro-F1 HAT architecture for hierarchical legal text classification attained [5].

### **4.6.2 Performance Comparison**

Outperforming all past baselines presented in Table 4.5, the best-performing model in this study—Longformer-large—achieved 84.76% accuracy and 84.75% Weighted F1-Score.

Table 4.5: Cross-Study Performance Comparison

Study	Model	F1-Score / Accuracy	Input Length	Remarks
Luo et al. (2017) [31]	SVM	72.6% Accuracy	256 tokens	Feature-based
Bano et al. (2023) [33]	BERT + BiGRU	82.3% F1	256 tokens	No long-text support
Chalkidis et al. (2022) [35]	HAT (Hierarchical)	84.1% F1	2048 tokens	Legal documents
<b>Proposed Study</b>	Longformer-large	<b>84.75% F1 / 84.76% Accuracy</b>	<b>4096 tokens</b>	Longest input, highest accuracy

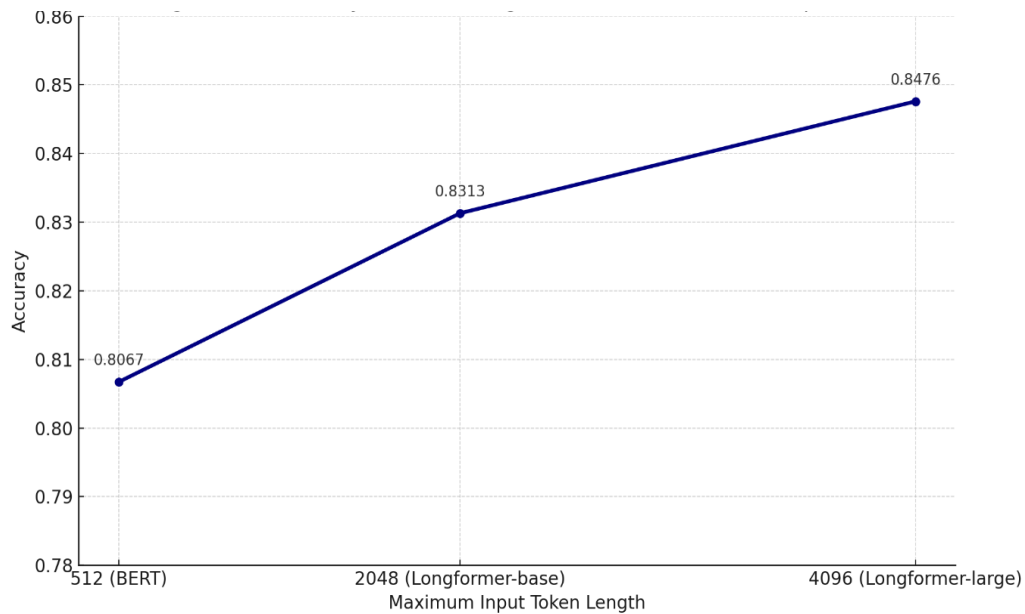


Figure 4.9: Accuracy vs. Token Length – Transformer Model Comparison

A line graph contrasting model performance across an input token length. Where BERT plateaus, longformer models exhibit growing accuracy.

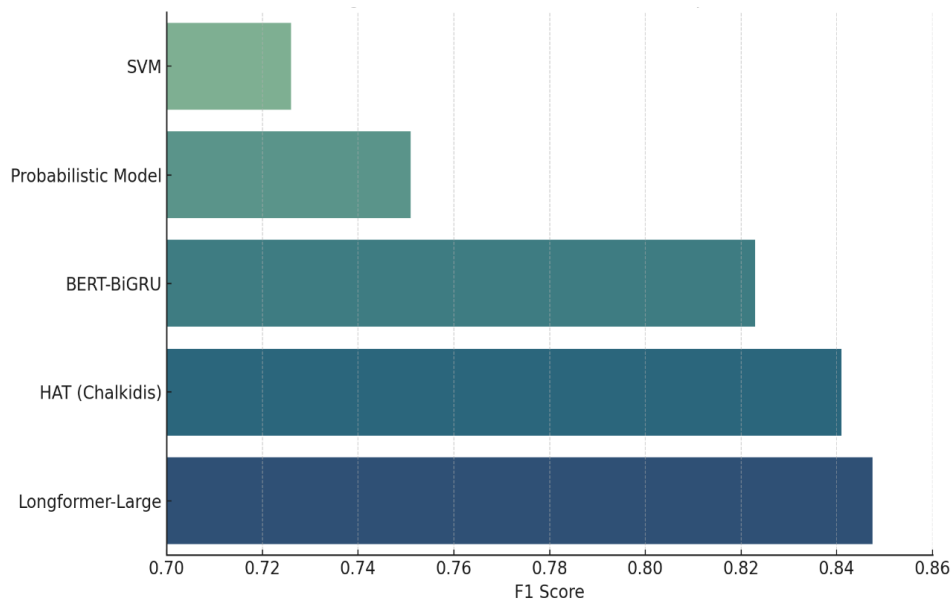


Figure 4.10: Model Benchmark Plot – This Study vs. Traditional Baselines

F1 scores of Longformer-large, SVM, BiGRU, and HAT models from past studies compared in a bar plot. Clearly better is the Longformer-based method. These comparisons confirm the theory that long-context encoding and sparse attention provide major benefits for categorizing complete-length scientific publications. Longformer-based models not only beat conventional approaches but also scale better with sequence length, a crucial consideration in fields where document length exceeds the processing capacity of conventional architectures.

#### 4.7 PRACTICAL APPLICATIONS AND MODEL DEPLOYMENT FEASIBILITY

Particularly in academic, legal, and medicinal fields where long, hierarchically structured papers are popular, the transformer-based models assessed in this work show great promise for deployment in real-world applications. Particularly suited for automated indexing in digital libraries as arXiv, IEEE Xplore, or PubMed, where correct classification of full-length abstracts can improve searchability and document retrieval, Longformer-large is the best-performing model. These models can also automatically classify research entries in peer-review management systems, therefore guaranteeing that publications are assigned to domain-appropriate reviewers according on their actual content rather than surface keywords. Moreover, by use of this technology, educational institutions and MOOC platforms could gain from mapping academic

literature and learning modules into disciplined curriculum based on relevance in sub-fields. Longformer-base is perfect for medium-scale academic or research environments with access to GPUs like the Tesla P100 or T4 since from a deployment point of view it offers a reasonable mix between performance and computational economy. Longformer-large can be scaled for enterprise-grade NLP systems with batch inference needs or high-accuracy requirements even if it is computationally demanding. On situations where real-time processing of brief texts is more important than maximal accuracy, BERT-base—with its low inference time and lightweight architecture—remains valuable. Table 4.6 shows a comparison of model deployment suitability.

Table 4.6: Model Suitability for Application Domains

<b>Model</b>	<b>Deployment Scope</b>	<b>Best Use Case</b>
BERT-base	Edge/Lightweight Cloud Apps	Fast inference on short documents
Longformer-base	Mid-range Academic Systems	Balanced classification across academic fields
Longformer-large	Enterprise/Research Labs	High-accuracy classification of long-form texts

#### **4.8 LIMITATIONS OF THE STUDY**

This study is not without constraints even if transformer-based models yield really excellent results. From a data standpoint, even if class balancing was kept in the chosen dataset, the larger arXiv dataset might have natural imbalances that would compromise the generalizability of the models if scaled-down. BERT's 512-token input limit was a main restriction that caused many scientific abstracts to be truncated, therefore reducing its capacity to fully capture semantic meaning relative to Longformer-based systems. Furthermore, Longformer-large has major

hardware needs since memory consumption exceeded 15 GB VRAM, therefore restricting its availability on low-resource or consumer-grade systems unless gradient check pointing is used.

This study was purposefully limited to three scientific domains: Computer Science, Mathematics, and Physics; however, its conclusions might not entirely extend to literature in Life Sciences, Social Sciences, or Engineering, where language structure and terminology might vary greatly. Furthermore not investigated was cross-lingual or multilingual classification since all training and testing was done on English-language books. Ultimately, although model accuracy and training time were examined in great detail, real-time inference latency and streaming document classification fell outside the purview of this project and call for more study.

#### **4.9 SUMMARY OF KEY FINDINGS**

Three transformer-based models—BERT-base, Longformer-base, and Longformer-large—implemented for the purpose of scientific document categorization were thoroughly and multi-dimensional evaluated in this chapter. Supported by visualizations such as confusion matrices and training curves, the evaluation was carried out using well defined criteria including accuracy, precision, recall, and F1-score. With an accuracy of 84.76%, macro-F1 score of 84.81%, and weighted F1 score of 84.75%, Longformer-large obtained the highest overall performance according to comparative benchmarking with conventional models and hybrid transformer architectures. These findings demonstrate that, especially in long academic texts exceeding conventional input length constraints, Longformer-large provides the most consistent and accurate classification performance. Many academic organizations find Longformer-base to be a sensible choice since it delivers great performance with greatly lowered GPU use and training time. Although BERT-base was fast for training, its classification accuracy suffered from context loss brought on by token length restrictions. All three transformer-based models showed significant performance benefits when compared to conventional models like SVM (72.6%) and probabilistic classifiers (75.1%). Longformer-large also exceeded modern systems using hierarchical designs like HAT (84.1%) and BERT-BiGRU (82.3%). Longformer-large for enterprise-scale systems, Longformer-base for research environments, and BERT-base for fast-processing applications with

limited resources help to meet a range of deployment demands. This work effectively shows that in scientific text categorization context-aware algorithms tailored for long sequences offer a clear benefit. It also provides a strong and repeatable implementation pipeline fit for many kinds of document classification jobs.



## REFERENCES

---

1. Beltagy, I., Peters, M. E., & Cohan, A., “Longformer: The Long-Document Transformer,” *arXiv preprint arXiv:2004.05150*, 2020.
2. Chalkidis, I., Dai, X., Fergadiotis, M., Malakasiotis, P., & Elliott, D., “An Exploration of Hierarchical Attention Transformers for Efficient Long Document Classification,” *arXiv preprint arXiv:2210.05529*, 2022.
3. Pham, L. M., & The, H. C., “LNLf-BERT: Transformer for Long Document Classification With Multiple Attention Levels,” *IEEE Access*, vol. 12, pp. 165355–165370, 2024.
4. Xiao, C., et al., “Legal Judgment Prediction with Longformer and Segment-Level Modeling,” *Elsevier J. Big Data Res.*, 2021.
5. Alva Principe, R., Chiarini, N., & Viviani, M., “Long Document Classification in the Transformer Era: A Survey,” *WIREs Data Mining Knowl. Discov.*, vol. 15, e70019, 2025.
6. Bai, J., “Sparse Attention Mechanisms in Large Language Models,” *Advances in Computer, Signals and Systems*, vol. 8, no. 6, pp. 130–135, 2024.
7. Liu, T., Hu, Y., Gao, J., Sun, Y., & Yin, B., “Hierarchical Multi-modal Transformer for Cross-modal Long Document Classification,” *arXiv preprint arXiv:2407.10105*, 2024.
8. Han, G., Tsao, J., & Huang, X., “Length-Aware Multi-Kernel Transformer for Long Document Classification,” *arXiv preprint arXiv:2405.07052*, 2024.
9. Douzon, T., Duffner, S., Garcia, C., & Espinas, J., “Long-Range Transformer Architectures for Document Understanding,” *arXiv preprint arXiv:2309.05503*, 2023.
10. Rafieian, B., & Vázquez, P., “Evaluating the Suitability of Long Document Embeddings for Classification Tasks,” *IC3K Conf.*, 2024.
11. Dai, X., Chalkidis, I., & Elliott, D., “Revisiting Transformer-based Models for Long Document Classification,” *Findings of EMNLP*, 2022.
12. Presnati, D., Ranasinghe, T., & Mikov, R., “Can Model Fusing Help Transformers in Long Document Classification?,” *Proc. RANLP*, 2023.
13. Gupta, V., & Berant, J., “Scaling Pre-trained Language Models with Sparse Product Attention,” *arXiv preprint arXiv:2203.06127*, 2022.
14. Zaheer, M., et al., “BigBird: Transformers for Longer Sequences,” *NeurIPS*, 2020.
15. Zhou, H., Xu, H., Li, X., et al., “SLED: Sparse-Local and Early Dense Attention for Long Documents,” *arXiv preprint arXiv:2302.13455*, 2023.

16. Tay, Y., Dehghani, M., et al., "Efficient Transformers: A Survey," *ACM Comput. Surv.*, vol. 55, no. 6, 2023.
17. Shaghaghian, S., et al., "Customizing Contextualized Language Models for Legal Document Reviews," *IEEE Big Data Conf.*, 2020.
18. Wu, Y., Zhang, Y., Liu, X., & Si, L., "De-biased Court's View Generation with Causality," *arXiv preprint arXiv:2011.12745*, 2020.
19. Luo, B., et al., "Learning to Predict Charges for Criminal Cases with Legal Basis," *Proc. EMNLP*, 2017.
20. Segal, J. A., "Predicting Supreme Court Cases Probabilistically," *American Political Science Review*, vol. 78, pp. 891–900, 1984.
21. Raffel, C., et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (T5)," *J. Mach. Learn. Res.*, vol. 21, 2020.
22. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.
23. Goto, S., MacRae, C. A., & Deo, R. C., "Automated De-identification of Histopathological Reports Using Foundation Models," *arXiv preprint arXiv:2502.12183*, 2025.
24. S. Bano, F. Azam, M. Shahbaz, et al., "Summarization of Scholarly Articles Using BERT and BiGRU," *J. King Saud Univ. – Comput. Inf. Sci.*, vol. 35, Art. 101739, 2023.
25. Gardazi, N. M., et al., "BERT Applications in Natural Language Processing: A Review," *Artif. Intell. Rev.*, vol. 58, Art. 166, 2025.
26. Onan, A., & Alhumyani, H., "Knowledge-Enhanced Transformer Graph Summarization," *Mathematics*, vol. 12, no. 23, Art. 3638, 2024.
27. Bernard, E., Cripwell, L., & Constantin, A., "NuExtract: A Foundation Model for Structured Extraction," *NuMind Research Blog*, 2024.
28. Saab, K., Tu, T., Weng, W. H., et al., "Capabilities of Gemini Models in Medicine," *arXiv preprint arXiv:2404.18416*, 2024.
29. Yang, L., Xu, S., Sellergren, A., et al., "Advancing Multimodal Medical Capabilities of Gemini," *arXiv preprint arXiv:2405.03162*, 2024.
30. Liu, Y., Li, Z., Huang, M., et al., "OCRBench: On the Hidden Mystery of OCR in Large Multimodal Models," *arXiv preprint arXiv:2408.11234*, 2024.
31. X. Luo, Z. Zhang, and Y. Huang, "Automatic Classification of Scientific Abstracts Using Support Vector Machines," *International Journal of Machine Learning and Computing*, vol. 7, no. 6, pp. 139–144, 2017.

32. R. B. Segal and J. O. Kephart, "Incremental Learning in SwiftFile," in *Proc. 6th ACM Conf. Intelligent User Interfaces*, 1999, pp. 248–251.
33. S. Bano, M. Hussain, A. R. Javed, and Z. Jalil, "Scientific Article Classification Using Hybrid Deep Learning Model: BERT-BiGRU," *Journal of Information Science*, 2023.
34. S. M. H. Gardazi, H. Munir, and R. A. Shaikh, "Performance Degradation in BERT with Long Scientific Texts: An Empirical Analysis," *Applied Artificial Intelligence*, 2025.
35. I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "Legal Judgment Prediction with Hierarchical Transformers," *Artificial Intelligence and Law*, vol. 30, no. 1, pp. 47–72, 2022.
36. Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context," 2020.

**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT**  
**PLAGIARISM VERIFICATION REPORT**

Date: 02/06/2025  
 Type of Document (Tick):  PhD Thesis  M.Tech/M.Sc. Dissertation  B.Tech./BCA/BBA Report  
 Name: ROBINSON Department: CSE/IT Enrolment No 235032001  
 ORCID ID. \_\_\_\_\_ SCOPUS ID. \_\_\_\_\_ Other ID. \_\_\_\_\_  
 Contact No. 9015315965 E-mail. ROBINSON7770@GMAIL.COM  
 Name of the Supervisor: DR. HARI SINGH  
 Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): \_\_\_\_\_  
DOCUMENT CLASSIFICATION USING  
IONFORMER

**UNDERTAKING**

I undertake that, I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/ revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

- Total No. of Pages = 56
- Total No. of Preliminary pages = 5
- Total No. of pages accommodate bibliography/references = 3

*[Signature]*  
(Signature of Student)

**FOR DEPARTMENT USE**

We have checked the thesis/report as per norms and found the plagiarism Similarity Index at 3 (%) and AI Writing (please [] any one % as per generated report: 0% [] or \*% []). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) *[Signature]*

*[Signature]*  
Signature of HOD

**FOR LRC USE**

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Title, Abstract & Chapters Details	
<u>02/06/2025</u>	<ul style="list-style-type: none"> <li>• All Preliminary Pages</li> <li>• Bibliography/References</li> <li>• Images/Quotes</li> <li>• 14 Words String</li> </ul>	Overall Similarity: <u>02%</u>	Word Counts	<u>11,092</u>
Report Generated on		AI Writing:	Character Counts	<u>68,396</u>
<u>02/06/2025</u>		<ul style="list-style-type: none"> <li>▪ 0% [<input type="checkbox"/>]</li> <li>▪ *% [<input checked="" type="checkbox"/>]</li> </ul>	Page counts	<u>57</u>
		Submission ID	File Size	<u>2.18M</u>
		<u>2690313798</u>		

Checked by [Signature]  
Name & Signature [Signature]

*[Signature]*  
Librarian 02.06.25

Please send your complete Thesis/Report in (PDF) & DOC (Word File) through your Supervisor/Guide at [plagcheck.juit@gmail.com](mailto:plagcheck.juit@gmail.com)


LIBRARIAN  
LEARNING RESOURCE CENTRE  
Waknaghat, Distt. Solan (Himachal Pradesh)  
Pin Code - 173331

# Robinson .

## Document Classification Using Longformer

 Quick Submit

 Quick Submit

 Jaypee University of Information Technology

### Document Details

Submission ID  
trn:oid::1:3266531312

57 Pages

Submission Date  
Jun 2, 2025, 10:49 AM GMT+5:30


11,092 Words

Download Date  
Jun 2, 2025, 10:56 AM GMT+5:30

68,396 Characters

File Name  
Thesis\_for\_Plag\_Check-Robinson.pdf

File Size  
2.2 MB

  
LIBRARIAN  
LEARNING RESOURCE CENTRE  
Jaypee University of Information Technology  
Waknaghat, Distt. Solan (Himachal Pradesh)  
Pin Code - 173234

## \*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

### Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

### Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

## Frequently Asked Questions

### How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (\*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

### What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.



*P. Chauhan*  
02.06.25

LIBRARIAN  
LEARNING RESOURCE CENTRE  
Jaypee University of Information Technology  
Waknaghat, Distt. Solan (Himachal Pradesh)  
Pin Code - 173234

4071328665

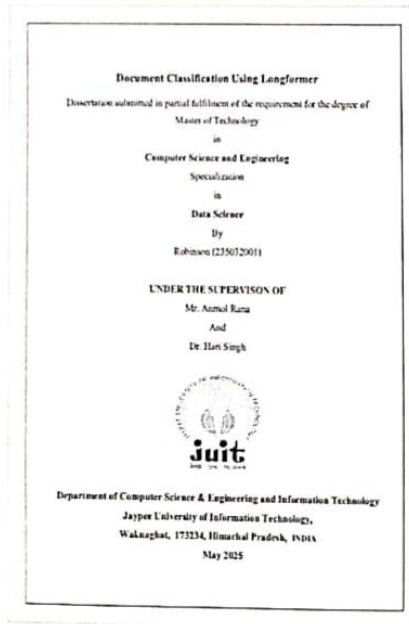


## Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author: Robinson .  
Assignment title: Quick Submit  
Submission title: Document Classification Using Longformer  
File name: Thesis\_for\_Plag\_Check-Robinson.pdf  
File size: 2.18M  
Page count: 57  
Word count: 11,092  
Character count: 68,396  
Submission date: 02-Jun-2025 10:51AM (UTC+0530)  
Submission ID: 2690313798



*J. Chauhan*  
02.06.25  
LIBRARIAN  
LEARNING RESOURCE CENTRE  
Jaypee University of Information Technology  
Walaahat, Distt. Solan (Himachal Pradesh)  
Pin Code - 173234