الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
جامعة غرداية
University of Ghardaia
كلية العلوم والتكنولوجيا
Faculty of Science and Technology
قسم الرياضيات والإعلام الآلي
Department of Mathematics and Computer Science
مخبر الرياضيات والعلوم التطبيقية
Mathematics and Applied Sciences Laboratory

Registration n°:
...../...../...../.....

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master

Domain: Mathematics and Computer
Field: Computer Science
Specialty: Intelligent Systems for Knowledge Extraction

Topic

Hybrid Learning for AI-Generated Text Detection and Attribution

Presented by

Asma Bensania & *Malak Dahma*

Publicly defended on June 22, 2026

Jury members

Dr Slimane Oulad-Naoui	MCA	Univ. Ghardaia	President
Dr Abdelfateh Bekkair	MAB	Univ. Ghardaia	Examiner
Dr Slimane Bellaouar	MCA	Univ. Ghardaia	Supervisor
Ms Kaoutar Zita	PhD Student	Univ. Ghardaia	Co-Supervisor

Academic Year: 2025/2026

Acknowledgment

First and foremost, we express our deepest gratitude to Allah, the Most Merciful and Compassionate, for His infinite blessings, guidance, and strength throughout every stage of this journey. It is by His grace and will that we have reached this significant milestone.

We sincerely thank our supervisors, ***Dr. Slimane Bellaouar*** and ***Ms. Kaoutar Zita***, for their exceptional mentorship, continuous support, and insightful guidance. Their expertise has been fundamental in shaping this research, refining our ideas, and maintaining our focus throughout this work. We are truly honored to have had the opportunity to work under their supervision.

We would also like to express our appreciation to the members of the jury for agreeing to review this work and for their constructive feedback and insightful remarks, which have helped enhance the quality of this thesis.

Special thanks are extended to the faculty members, administrative staff, and colleagues at the University of Ghardaïa for creating a supportive and stimulating academic environment throughout our studies.

Finally, we extend our gratitude to all those who contributed in any way to the completion of this thesis.

May Allah reward you all for their kindness and support.

Dedication

First and foremost, I would like to thank Allah, whose infinite mercy, guidance, and blessings have been the source of strength and perseverance throughout this work.

I dedicate this achievement to my beloved **parents**, whose love, sacrifices, and prayers have always guided my path. I am deeply grateful for their constant support, patience, and unwavering faith, which made this accomplishment possible.

To my dear siblings, **Aya**, **Islam**, and **Oussama**, thank you for your continuous encouragement, understanding, and support throughout this journey.

To my **husband**, I sincerely thank you for your constant support, patience, and encouragement. Your presence and understanding gave me strength during the most challenging moments of this academic journey.

To my dear friend and project partner, **Malak**, through every stage of this journey, you stood by my side. Your collaboration, commitment, and support are reflected in every step of this work, and I am truly grateful for your presence throughout this experience.

Finally, I dedicate this work to everyone who believed in me, supported me, and encouraged me throughout this journey. May this achievement stand as a humble reflection of my deep gratitude to all those who stood by my side.

Bensania Assma

Dedication

Above all, I am deeply grateful to Allah, whose infinite mercy, guidance, and blessings gave me the strength and perseverance to complete this work.

I dedicate this work to my beloved **parents**, whose unconditional love, sacrifices, and prayers have always been the light guiding my path. Their unwavering support and faith made this accomplishment possible.

I dedicate it to my dear **grandfather** and **grandmother**, whose wisdom, love, prayers, and inspiration have always been a source of strength and motivation.

To my dear siblings, **Fouad**, **Nouha**, and **Djoud**, thank you for your continuous encouragement, understanding, and support throughout this journey.

To my dear friend and project partner, **Asma**, and my dear friend **Aya**, thank you for your friendship, encouragement, and support. Your presence made this journey more meaningful and memorable. A special thank you to **Wissal** for her trust in me and her constant encouragement.

Finally, I dedicate this work to everyone who believed in me, supported me, and encouraged me throughout this journey, and to myself for the perseverance, resilience, and dedication invested in reaching this milestone. May this achievement stand as a humble reflection of my gratitude to all those who stood by my side.

Dahma Malak

الملخص

أدى التطور السريع للنماذج اللغوية الضخمة (LLMs) إلى تحسين جودة وطلاقة النصوص المولدة بالذكاء الاصطناعي بشكل كبير، مما جعل التمييز بينها وبين النصوص البشرية أكثر صعوبة. وتتزايد هذه الصعوبة في مهام إسناد المصدر، حيث تُظهر النصوص المولدة بواسطة عدة نماذج لغوية خصائص لغوية وأسلوبية متداخلة بشكل كبير. تقترح هذه الدراسة إطارًا هجينًا لاكتشاف النصوص المولدة بالذكاء الاصطناعي وإسناد مصدرها من خلال دمج التمثيلات الدلالية والأسلوبية بهدف تحسين أداء التصنيف. يعتمد الإطار المقترح على تضمينات دلالية سياقية مستخرجة باستخدام RoBERTa مع خصائص أسلوبية مصممة يدويًا، بالإضافة إلى استخدام التعلم التبايني الموجَّه لتحسين قابلية تمييز الخصائص والفصل بين الفئات. يعالج النهج المقترح كلًا من التصنيف الثنائي (نص بشري مقابل نص مولد بالذكاء الاصطناعي) والإسناد متعدد الفئات لتحديد النموذج المولد للنص. بعد ذلك، تتم معالجة التمثيل المدمج باستخدام مصنف متعدد الطبقات (MLP) لإجراء التنبؤ النهائي. أظهرت التجارب المنجزة على مجموعة بيانات MAGE فعالية الإطار المقترح، حيث حقق نسبة Macro F1 بلغت 93.20% في التصنيف الثنائي و82.22% كـ Macro F1 في الإسناد متعدد الفئات. كما بينت النتائج المقارنة تفوق النموذج المقترح على عدة نماذج تقليدية للتعلم الآلي، من بينها Logistic Regression وRandom Forest وSVM، مما يؤكد متانة وفعالية النهج المقترح في مهام اكتشاف النصوص المولدة بالذكاء الاصطناعي وإسناد مصدرها.

الكلمات المفتاحية: اكتشاف النصوص المولدة بالذكاء الاصطناعي، النماذج اللغوية الضخمة (LLMs)، إسناد المصدر، التعلم التبايني.

Abstract

The rapid advancement of large language models (LLMs) has significantly increased the quality and fluency of AI-generated text, making it increasingly difficult to distinguish from human-written content. This challenge is further amplified in source attribution tasks, where texts generated by multiple LLMs often exhibit overlapping linguistic and stylistic characteristics. This study proposes a hybrid framework for AI-generated text detection and attribution by combining semantic and stylometric representations to improve classification performance. The framework integrates contextual semantic embeddings extracted using RoBERTa with handcrafted stylometric features, while supervised contrastive learning is employed to enhance feature discriminability and class separability. The proposed approach addresses both binary classification (human versus AI-generated text) and multi-class attribution for identifying the generating model. The fused representation is processed through a multilayer perceptron (MLP) classifier for final prediction. Experimental evaluations conducted on the MAGE Dataset demonstrate the effectiveness of the proposed framework, achieving a 93.20% macro F1-score in binary classification and an 82.22% macro F1-score in multi-class attribution. Comparative results show that the proposed model outperforms traditional machine learning baselines, including Logistic Regression, Random Forest, Support Vector Machine (SVM), demonstrating its robustness and effectiveness for AI-generated text detection and attribution tasks.

Keywords: AI-generated text detection, large language models (LLMs), source attribution, contrastive learning.

Résumé

L'avancement rapide des grands modèles de langage (LLMs) a considérablement amélioré la qualité et la fluidité des textes générés par intelligence artificielle, rendant leur distinction par rapport aux textes rédigés par des humains de plus en plus difficile. Ce défi est encore accentué dans les tâches d'attribution de source, où les textes générés par plusieurs LLMs présentent souvent des caractéristiques linguistiques et stylistiques fortement chevauchantes. Cette étude propose un cadre hybride pour la détection et l'attribution des textes générés par l'IA, en combinant des représentations sémantiques et stylométriques afin d'améliorer les performances de classification. Le cadre intègre des embeddings sémantiques contextuels extraits à l'aide de RoBERTa avec des caractéristiques stylométriques conçues manuellement, tandis que l'apprentissage contrastif supervisé est utilisé pour améliorer la séparabilité des caractéristiques et des classes. L'approche proposée traite à la fois la classification binaire (texte humain contre texte généré par IA) et l'attribution multi-classes pour identifier le modèle générateur. La représentation fusionnée est ensuite exploitée par un classifieur de type perceptron multicouche (MLP) pour la prédiction finale. Les évaluations expérimentales menées sur le jeu de données MAGE montrent l'efficacité de l'approche proposée, avec un score F1 macro de 93.20% en classification binaire et un score F1 macro de 82.22 % en attribution multi-classes. Les résultats comparatifs montrent que le modèle proposé surpasse plusieurs méthodes d'apprentissage automatique classiques, notamment la régression logistique, les forêts aléatoires, les machines à vecteurs de support (SVM), démontrant ainsi sa robustesse et son efficacité pour les tâches de détection et d'attribution de textes générés par l'IA.

Mots-clés : détection de texte généré par l'IA, grands modèles de langage (LLMs), attribution de source, apprentissage contrastif.

Contents

List of Figures

List of Tables

Introduction

The rapid advancement of large language models (LLMs) has significantly transformed the field of natural language processing (NLP), enabling the generation of text that is increasingly fluent, coherent, and often indistinguishable from human-authored content. Transformer-based architectures, particularly GPT-style models and their variants, have demonstrated strong performance across a wide range of tasks, including text generation, machine translation, summarization, and conversational systems [Brown et al., 2020]. Despite these remarkable capabilities, the growing sophistication of LLMs raises critical concerns regarding the authenticity, reliability, and potential misuse of automatically generated content. In particular, modern LLMs are capable of producing highly human-like text, which makes distinguishing between human-written and machine-generated content increasingly challenging [Zellers et al., 2019]. Consequently, AI-generated text detection has emerged as an important problem in NLP, commonly formulated as a binary classification task aimed at discriminating between human and AI-generated text [Solaiman et al., 2019].

Existing methods rely heavily on surface-level or semantic features alone, which may not be sufficient to capture subtle differences in writing patterns. Furthermore, limited attention has been given to jointly addressing both detection and attribution, where identifying the specific source model of generated text remains an open challenge. These limitations highlight the need for more robust and unified approaches capable of handling both detection and attribution tasks effectively.

To address these limitations, this work proposes a hybrid framework for AI-

generated text detection and source attribution, combining binary classification (human vs AI) with multi-class LLM identification. The proposed approach leverages transformer-based semantic representations to capture contextual information in text, complemented by stylometric features that reflect writing style characteristics. To further enhance representation quality and improve class separability, supervised contrastive learning is employed during training. This unified framework enables the model to simultaneously perform detection and attribution, providing a more comprehensive solution to AI-generated text analysis.

This thesis is organized as follows. Chapter 1 introduces the fundamental concepts of natural language processing and deep learning, including recurrent neural networks (RNNs), long short-term memory (LSTM) networks, transformer architectures, and large language models (LLMs) such as GPT, T5, and LLaMA. Chapter 2 presents a structured review of existing approaches for AI-generated text detection, covering statistical methods, machine learning techniques, deep learning models, and hybrid approaches, as well as benchmark datasets used in this field. Chapter 3 describes the proposed methodology, including dataset preparation, feature extraction, and the overall system architecture. Chapter 4 details the experimental setup and results analysis, including the hyperparameter configuration, evaluation protocol, and performance comparison under both binary classification and multi-class attribution scenarios. Finally, Chapter 4.4 presents the conclusion and future perspectives of this work.

Chapter 1

Basic Concepts

1.1 Introduction

The development of Large Language Models (LLMs) has enabled AI to generate highly fluent and coherent text, creating challenges in distinguishing machine-generated content from human writing. This chapter presents the foundational concepts underlying this research, including text representation techniques, representation learning paradigms, traditional machine learning and deep learning models for textual data processing, as well as an overview of LLMs and AI-generated text detection, its applications, and associated challenges.

1.2 Text Representation Techniques

Text representation is a fundamental step in natural language processing that converts raw textual data into structured numerical forms suitable for machine learning models. This section presents the main representation approaches, from classical sparse methods to modern contextual embeddings.

1.2.1 Classical Methods

Classical text representation methods convert documents into numerical vectors based on word occurrence and frequency statistics, with **Bag-of-Words** and **TF-IDF** as two representative techniques.

Bag of Words

Bag-of-Words (BoW) is a text representation method that models a document as a vector over a fixed vocabulary. Each word is encoded as 1 if it appears in the document and 0 otherwise, or by its frequency. This results in a document-term matrix of size $|\text{documents}| \times |V|$, where |V| is the number of unique terms in the vocabulary. However, the BoW ignores word order and semantic relationships, focusing only on word occurrence[Harris, 1954, Patil et al., 2023].

TF-IDF

Term Frequency–Inverse Document Frequency is a statistical text representation technique introduced by Jones [2004] that measures the importance of a word in a document relative to a corpus. It assigns higher weights to words that frequently appear in a document but rarely occur across the corpus [Jones, 2004, Havrlant and Kreinovich, 2017].

The TF-IDF score is computed as:

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t)$$

where:

$$\text{TF}(t,d) = \frac{f_{t,d}}{\sum_k f_{k,d}}$$

$$\text{IDF}(t) = \log\left(\frac{N}{1+n_t}\right)$$

Here, $f_{t,d}$ is the frequency of term t in document d, N is the total number of

documents, and n_t is the number of documents containing term t [Patil et al., 2023].

1.2.2 Word Embeddings

Word embeddings are dense vector representations of words that capture semantic relationships in a continuous vector space. Popular models such as Word2Vec [Mikolov et al., 2013], GloVe [Pennington et al., 2014], and FastText [Bojanowski et al., 2017] learn these representations from large text corpora using contextual or co-occurrence information. However, these models produce static representations by assigning a single fixed vector to each word, regardless of its contextual usage. This limitation restricts their ability to model polysemy and capture context-dependent meanings [Patil et al., 2023].

1.2.3 Contextual Representations

Contextual embeddings are dynamic word representations in which the meaning of a word is determined according to its surrounding context. Unlike static embeddings, the same word can have different vector representations depending on its usage in a sentence. Models such as **BERT** [Devlin et al., 2019] use Transformer architectures and self-attention mechanisms to capture bidirectional contextual information from text. These embeddings have significantly improved the performance of many natural language processing tasks such as text classification, sentiment analysis, and named entity recognition [Patil et al., 2023].

1.3 Representation Learning Paradigms

Representation learning aims to extract structured and informative feature representations from raw data, enabling machine learning models to perform effectively across a wide range of tasks. This section presents the principal learning paradigms

that underpin modern representation learning, namely supervised learning, self-supervised learning, and contrastive learning.

1.3.1 Supervised Learning

Supervised learning is a machine learning paradigm in which a model learns the relationship between input data and corresponding target labels using labeled training samples. The objective is to learn a mapping that enables the prediction of outputs for unseen data. During training, the model generates predictions that are compared with the true labels, and the resulting error is used to optimize the model parameters and improve performance. Supervised learning is widely used for classification and regression tasks due to its effectiveness and ability to produce accurate and interpretable predictions. It has been successfully applied in various domains, including natural language processing, computer vision, speech recognition, and spam detection. However, supervised learning strongly depends on the availability of large, high-quality labeled datasets, whose collection and annotation can be costly and time-consuming. In addition, supervised models may suffer from overfitting, leading to poor generalization on unseen data [Liu and Wu, 2012].

1.3.2 Unsupervised Learning

Unsupervised learning is a fundamental machine learning paradigm in which models learn directly from unlabeled data without requiring predefined target labels or manual annotation. Unlike supervised learning, the learning process is driven by the inherent characteristics of the data rather than explicit supervisory signals. The primary objective is to uncover latent structures, hidden patterns, and meaningful relationships embedded within the data, thereby enabling the automatic extraction of informative representations. Owing to its ability to exploit large volumes of unlabeled data, unsupervised learning has become an essential component of modern machine learning and is widely employed in tasks such as clustering, dimensionality reduction,

anomaly detection, and representation learning [Usama et al., 2019].

1.3.3 Self-Supervised Learning

Self-supervised learning (SSL) is a representation learning paradigm in deep learning in which models learn meaningful representations from unlabeled data by generating supervisory signals directly from the data itself. According to recent literature, SSL exploits input data as supervision through carefully designed **pretext** tasks that enable the model to predict missing or transformed parts of the input, thereby learning useful feature representations without manual annotation, unlike supervised learning. These pretext tasks may include predicting missing words, reconstructing corrupted inputs, or modeling relationships within the data structure. SSL methods are generally categorized into generative and contrastive approaches, depending on whether the model learns by reconstructing the input data or by distinguishing between similar and dissimilar representations. In natural language processing, this paradigm is widely used in models such as **BERT** [Devlin et al., 2019] through masked language modeling and **GPT** [Radford and Narasimhan, 2018] through next-token prediction, where the learning signal is automatically derived from raw textual data [de Sa, 1993, Liu et al., 2021] .

1.3.4 Contrastive Learning

Contrastive learning is a representation learning framework that structures an embedding space by pulling semantically similar samples closer together while pushing dissimilar ones farther apart, without relying on explicit class annotations [Hadsell et al., 2006]. This process is illustrated in **Figure 1.1**. The framework is optimized via a loss function defined over labeled pairs $(\vec{X}_1, \vec{X}_2)$ with a binary label $Y \in \{0, 1\}$, where $Y = 0$ indicates a similar pair and $Y = 1$ indicates a dissimilar pair. The classical margin-based contrastive loss, introduced by Hadsell et al. [2006], is defined

as:

$$\mathcal{L}(W, Y, \vec{X}_1, \vec{X}_2) = (1 - Y)\frac{1}{2} D_W^2 + Y\frac{1}{2}\left[\max(0, m - D_W)\right]^2$$

where $D_W = \|G_W(\vec{X}_1) - G_W(\vec{X}_2)\|_2$ is the Euclidean distance between the encoded representations, G_W is the encoder parameterized by learnable weights W, and $m > 0$ is a margin hyperparameter that enforces a minimum separation between dissimilar pairs [Hadsell et al., 2006]. Several formulations have since been proposed, including InfoNCE in MoCo [He et al., 2020] and NT-Xent in SimCLR [Chen et al., 2020], while supervised contrastive learning [Khosla et al., 2021] further extends the framework by incorporating class labels to define multiple positive pairs per anchor. In self-supervised settings, the model learns from unlabeled data by treating augmented views of the same sample as positive pairs, achieving strong performance in tasks such as image representation learning and sentence embedding, whereas in supervised settings, class labels are used to define positive pairs, yielding more discriminative representations for downstream tasks such as image recognition and text classification.

1.4 Machine Learning and Deep Learning

Modern artificial intelligence applications rely on a diverse set of learning models, spanning traditional machine learning and deep learning architectures. These methods are mainly categorized under supervised and self-supervised learning paradigms. In this section, we present several representative models from both approaches and their practical applications.

1.4.1 Traditional Machine Learning Models

Traditional machine learning includes supervised learning approaches designed to learn predictive patterns for classification and regression tasks. In this subsection, we

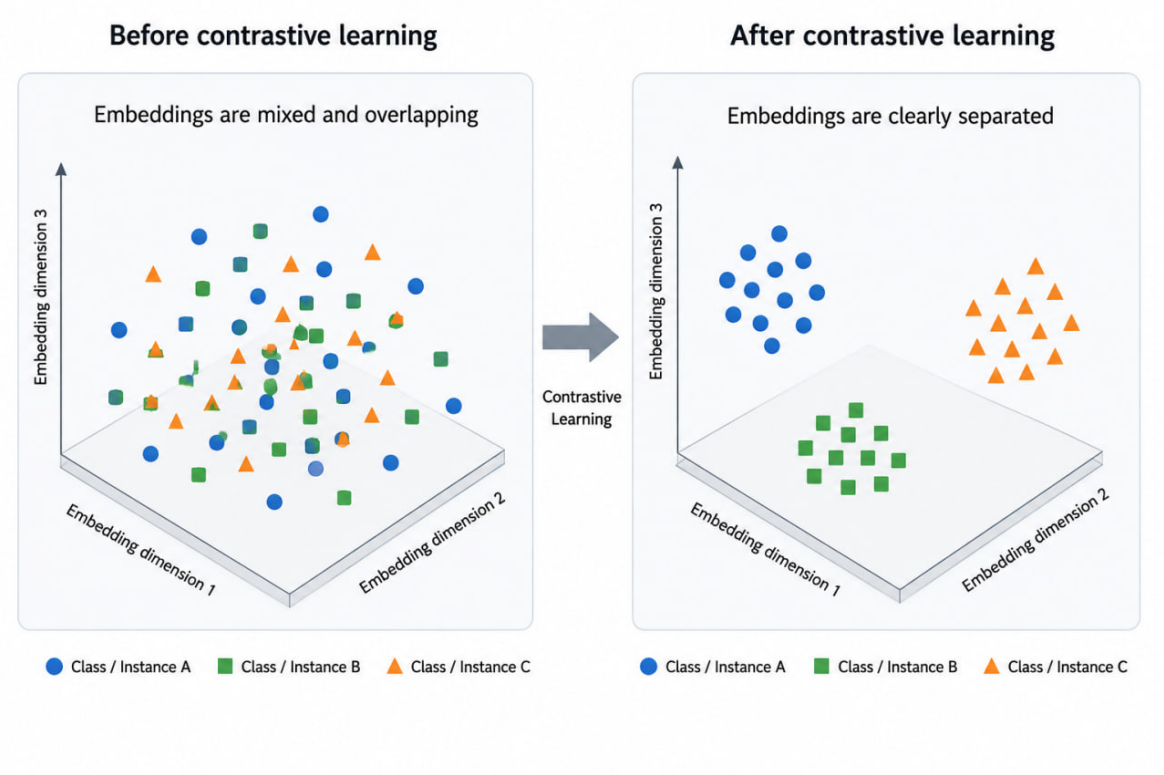


Figure 1.1: Visualization of contrastive learning in embedding space.

focus on classification models and present **Logistic Regression** and **Support Vector Machine** as two commonly used methods.

Logistic Regression

Logistic Regression is a statistical model commonly employed for binary classification tasks and can also be extended to multiclass classification problems, where the relationship between input features and a categorical dependent variable is modeled using the logistic sigmoid function to estimate class probabilities. Formally, given an input vector x, the predicted probability is estimated as:

$$P(y = 1 \mid x) = \sigma(w^T x + b) = \frac{1}{1 + e^{-(w^T x + b)}}$$

where w is the learnable weight vector, b is the bias term, and $\sigma(\cdot)$ denotes the logistic sigmoid function. The model learns an optimal decision boundary by optimizing its parameters through maximum likelihood estimation, assigning each instance to one

of two classes based on the estimated probability. Due to its interpretability and computational efficiency, it remains a well-established baseline for classification tasks across numerous application domains [Cucchiara, 2012].

Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm widely used for classification tasks, which seeks to determine an optimal separating hyperplane between data points belonging to different classes. The model operates by maximizing the margin between the decision boundary and the closest training samples, referred to as support vectors, thereby enhancing generalization performance. Furthermore, SVM can be extended to handle non-linear classification problems through the use of kernel functions, which implicitly project the input data into higher-dimensional feature spaces where linear separation becomes feasible [Cortes and Vapnik, 1995].

1.4.2 Deep Learning Models

Deep learning is a branch of machine learning based on multi-layer artificial neural networks that learn hierarchical representations from data. These methods are mainly used within supervised and self-supervised learning settings. This subsection presents key deep learning models designed to capture complex dependencies and structures in sequential data .

Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are neural architectures designed to process sequential data by maintaining an internal state that evolves over time. This recurrent structure enables the model to incorporate information from previous inputs when processing current elements in a sequence. The foundational formulation of this idea was introduced by Elman [1990], who proposed a simple recurrent network with contextual

feedback connections, allowing the model to capture temporal dependencies and learn structured patterns in sequential data .

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a specialized type of Recurrent Neural Network designed to model sequential data and capture long-range dependencies. Unlike standard RNNs, LSTMs are equipped with a memory cell that allows information to be stored, updated, or removed over time through a set of gating mechanisms. These gates control the flow of information and help the model retain relevant context over long sequences. This architecture was introduced to address the vanishing gradient problem, which limits the ability of traditional RNNs to learn long-term dependencies in sequential data [Hochreiter and Schmidhuber, 1997].

Figure 1.2 illustrates the structural differences between RNN and LSTM architectures.

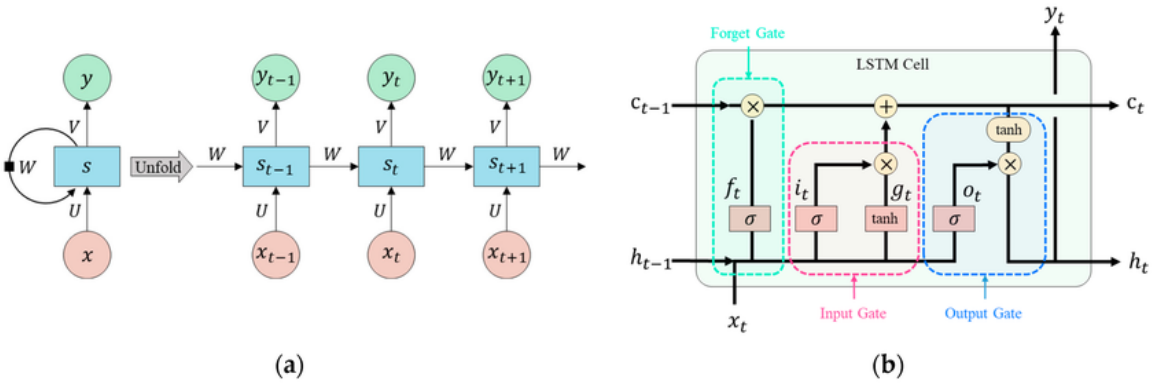


Figure 1.2: Overview of RNN and LSTM Architectures[Kim et al., 2023].

Transformer

The Transformer is a deep learning architecture designed for sequence modeling tasks using attention mechanisms to capture contextual relationships between words. As illustrated in Figure 1.3, it is composed of two main components: the encoder and the decoder. The encoder processes the input sequence and generates contextual

representations, while the decoder uses these representations along with previously generated outputs to produce the target sequence. Both the encoder and decoder are built from multi-head self-attention mechanisms and feed-forward neural networks combined with residual connections and layer normalization. The core mechanism of the Transformer is self-attention, which enables the model to learn dependencies between tokens through Query, Key, and Value representations, allowing efficient understanding of long-range contextual information within a sequence [Vaswani et al., 2023].

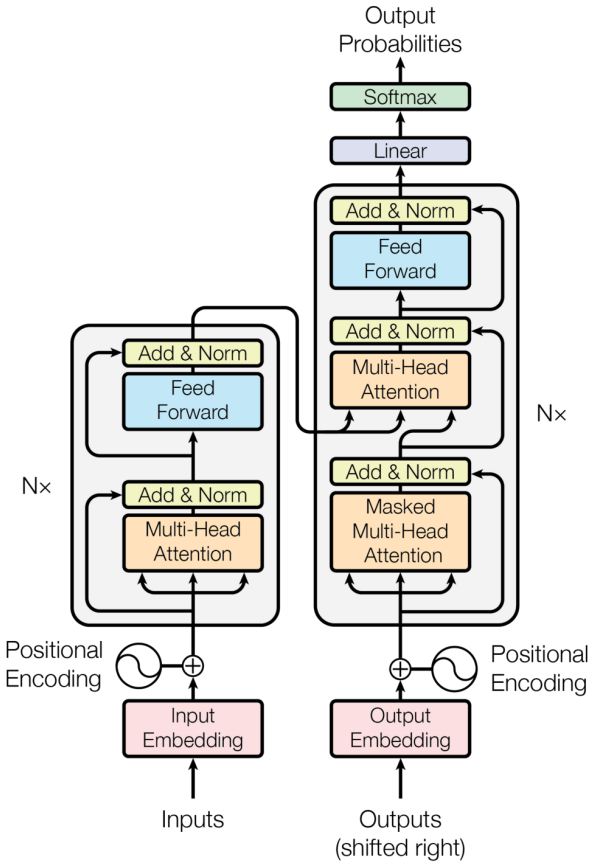


Figure 1.3: Illustration of the Transformer Architecture[Vaswani et al., 2023].

1.5 Large Language Models (LLMs)

Large Language Models (LLMs) are deep learning models trained on massive text corpora to understand and generate human-like language. These models leverage the Transformer architecture, which uses self-attention mechanisms to capture contextual

relationships across text sequences [Vaswani et al., 2023, Kasneci et al., 2023]. LLMs have demonstrated remarkable capabilities in generating fluent and coherent text, performing summarization, translation, question answering, and other natural language processing tasks. Table 1.1 below presents some of the most common generative Large Language Models (LLMs) used in natural language processing. In the following subsections, we focus on three of the most prominent models T5, GPT, and LLaMA.

Table 1.1: Common generative Large Language Models

Model	**Version**	**Developer**	**Parameters**	**Ref.**
GPT	GPT-3	OpenAI	175B	[Brown et al., 2020]
OPT	v1	Meta AI	125M–175B	[Zhang et al., 2022]
FLAN-T5	Small–XXL	Google	80M–11B	[Chung et al., 2022]
T5	Small–11B	Google	60M–11B	[Raffel et al., 2023]
BLOOM	BLOOM	BigScience / Hugging Face	176B	[Workshop et al., 2022]
LLaMA	LLaMA 1	Meta AI	7B–65B	[Touvron et al., 2023]
Gemini	1.0	Google DeepMind	Not publicly disclosed	[Gemini Team et al., 2024]
DeepSeek	DeepSeek LLM	DeepSeek AI	7B–67B	[DeepSeek-AI, 2024]
Claude	Claude 3	Anthropic	Not publicly disclosed	[Anthropic, 2024]

1.5.1 Text-to-Text Transfer Transformer (T5)

Text-to-Text Transfer Transformer (T5) is an artificial intelligence model designed to unify a wide range of natural language processing tasks under a single framework by representing every task as a text-to-text problem, where both inputs and outputs are sequences of text. T5 is built on an encoder-decoder Transformer architecture and is pre-trained on a large cleaned dataset of web text called the Colossal Clean Crawled Corpus (C4), using a span-corruption objective that enables it to learn robust general language representations.

During the pre-training phase, T5 learns to reconstruct missing spans of text based on surrounding context, allowing it to capture deep syntactic and semantic patterns. During inference, the model applies the same text-to-text formulation to a variety of downstream tasks, such as translation, summarization, question answering, and classification, by conditioning on appropriately formatted input prompts. This unified paradigm simplifies task handling and model deployment, enabling T5 to leverage transfer learning effectively across diverse Natural Language Processing

(NLP) benchmarks using a single, consistent architecture [Raffel et al., 2023].

1.5.2 Generative Pre-trained Transformer (GPT)

Generative Pre-trained Transformer (GPT-3) is an artificial intelligence language model designed to generate coherent and contextually relevant text by predicting the next token in a sequence. GPT-3 is based on an autoregressive Transformer architecture and was developed by OpenAI, featuring 175 billion parameters, which enables it to capture complex linguistic patterns from large-scale textual data.

GPT-3 relies on large-scale unsupervised pre-training on a diverse corpus of internet text to learn general language representations. Unlike traditional task-specific models, GPT-3 does not require fine-tuning for each downstream task. Instead, it adopts an in-context learning paradigm, where task descriptions and a small number of examples are provided directly within the input prompt.

The operational process of GPT-3 begins with Pre-training, during which the model learns to predict the next word based on preceding context using a language modeling objective. Following this, at inference time enables the model to perform various natural language processing tasks, such as translation, question answering, summarization, and text completion, by conditioning on input prompts. In the Few-shot, One-shot, or Zero-shot settings, GPT-3 adapts to new tasks solely through contextual examples, without modifying its internal parameters. This flexible workflow allows GPT-3 to generalize across multiple tasks using a single unified model architecture [Brown et al., 2020].

1.5.3 Large Language Model Meta AI (LLaMA)

LLaMA (Large Language Model Meta AI) is a family of foundation language models developed by Meta AI that aim to provide strong performance across a wide range of natural language tasks. The models are autoregressive Transformer-based

architectures with sizes ranging from 7 billion to 65 billion parameters. LLaMA models are trained on trillions of tokens sourced exclusively from publicly available datasets, including Common Crawl, C4, Wikipedia, Books, GitHub, ArXiv, and StackExchange, demonstrating that state-of-the-art results can be achieved without proprietary data.

The training process focuses on efficiency and performance. Optimized architectural designs and training techniques are used to improve learning and inference speed. Once trained, the models can perform diverse tasks such as text generation, summarization, question answering, and reasoning. Smaller models, like LLaMA-13B, are shown to outperform GPT-3 (175B) on most benchmarks, while larger models, like LLaMA-65B, remain competitive with models such as Chinchilla-70B and PaLM-540B, despite having fewer parameters.

All LLaMA models and weights are released to the research community. This enables researchers to conduct experiments, fine-tune the models for specific tasks, and ensure open access and reproducibility in large language model research [Touvron et al., 2023].

1.6 AI-Generated Text Detection

The widespread adoption of Large Language Models (LLMs) in various applications has enabled the generation of highly fluent and human-like text through large-scale pre-training and fine-tuning. However, this progress has also raised an important challenge of reliably distinguishing AI-generated text from human-written content .

In this section, we present the problem of AI-generated text detection, including the tasks of binary classification and multi-class attribution, and discuss its main applications and associated challenges.

1.6.1 Problem Definition

In this work, we address AI-generated text detection through two related tasks: binary classification and multi-class attribution.

Binary Classification. The objective is to determine whether a given text x is human-written or machine-generated. This problem is formulated as a function $D_b(x)$ that assigns a binary label to the input text.

$$D_b(x) = \begin{cases} 0, & \text{if } x \text{ is human-written,} \\ k,\ k \in \{1, \ldots, 6\}, & \text{if } x \text{ is AI-generated.} \end{cases}$$

Multi-Class Attribution. In addition to binary detection, the objective is to identify the source of a given text. This is formulated as a function $D_m(x)$ that assigns the input text to one of multiple possible classes:

$$D_m(x) = \begin{cases} 0, & \text{if } x \text{ is human-written} \\ 1, & \text{if } x \text{ is generated by } \mathrm{LLM}_1 \\ 2, & \text{if } x \text{ is generated by } \mathrm{LLM}_2 \\ \vdots & \\ K, & \text{if } x \text{ is generated by } \mathrm{LLM}_K \end{cases}$$

Binary classification and multi-class attribution address complementary aspects of the problem, the first distinguishes human-written from machine-generated text, while the second identifies the specific generation source. The goal is to develop a reliable approach for AI-generated text detection and source attribution across different language models.

1.6.2 Applications of AI-Generated Text Detection

The rapid development of large language models (LLMs) has led to the widespread generation of synthetic text across various domains. While these models provide significant benefits in terms of productivity and accessibility, they also raise concerns related to authenticity, reliability, and ethical use, making AI-generated text detection an important task for ensuring trust in text-based systems.

Education

In education, generative AI tools such as ChatGPT offer support for learning and teaching, but their misuse in academic work may affect students' skills and academic integrity, highlighting the need for reliable detection mechanisms [Grassini, 2023, Susnjak, 2022].

Medicine

In medicine, large language models can assist in clinical tasks and decision-making, but their outputs may contain inaccuracies, requiring careful validation before use in critical environments [Hajijama et al., 2024, Thirunavukarasu et al., 2023].

Coding

In programming, AI-generated code can assist with debugging and development tasks, but it may also introduce functional or security-related issues, requiring human verification before deployment [Feng et al., 2023, Khoury et al., 2023].

Law

In the legal domain, LLMs can generate coherent legal text and assist in information processing, but their reliability remains limited for professional use due to variations in accuracy [Lam et al., 2023, Tan et al., 2023].

News

In journalism, AI-generated content can support news production, but it may introduce bias and lack stylistic authenticity compared to human writing, motivating the need for detection systems [Wang et al., 2023, Fang et al., 2024].

1.6.3 Challenges in AI-Generated Text Detection

Detecting AI-generated text remains a challenging research problem due to several technical, linguistic, and ethical limitations. One major challenge lies in identifying AI-polished human writing, where human-authored texts are lightly edited or refined using AI tools. In such cases, detection systems often misclassify genuinely human content as AI-generated, leading to false positives and raising concerns about fairness and reliability in high-stakes applications such as academic integrity assessment [Saha and Feizi, 2025].

Another critical limitation arises in the detection of hybrid human–AI texts, where content is produced through collaboration between humans and generative language models. When authors selectively integrate or modify AI-generated sentences, the resulting text exhibits blended stylistic features that confuse existing detectors. Binary classification approaches struggle to determine authorship boundaries within such texts, significantly reducing detection accuracy [Zeng et al., 2024].

The increasing sophistication of generative language models such as GPT, LLaMA, and PaLM adds another layer of difficulty, as these models produce outputs that closely resemble human writing, making reliable detection inherently complex. Therefore, detection systems must be continuously updated to keep pace with the evolution of generative models [Fariello et al., 2024].

AI-generated text detectors are highly vulnerable to adversarial attacks. Research has demonstrated that small, intentional modifications such as synonym substitution or sentence restructuring can successfully evade detection. This susceptibility highlights

the lack of robustness in current approaches and emphasizes the need for more resilient and manipulation-aware detection frameworks [Zhou et al., 2024].

Finally, regulatory and ethical gaps remain a major challenge. The absence of standardized frameworks for transparency, labeling, and responsible deployment of AI-generated content limits the integration of detection systems in academic, commercial, and legal domains. These gaps complicate policy-making and raise concerns about accountability and fairness in AI content evaluation [Fariello et al., 2024].

Overall, these challenges indicate that AI-generated text detection is not only a technical problem but also a broader linguistic, ethical, and regulatory issue. Addressing these limitations requires adaptive models, continuous updates to keep pace with evolving generative models, robust defenses against adversarial manipulation, and the development of standardized guidelines for ethical deployment.

1.7 Conclusion

This chapter explored the fundamental concepts essential for understanding AI-generated text detection and the underlying technologies involved in this research. The chapter also reviewed the key techniques, models, and challenges associated with AI-generated text detection. In the next chapter, we examine in detail the state of the art in AI-generated text detection.

Chapter 2

State of the Art

2.1 Introduction

This chapter reviews related work on AI-generated text detection and source attribution. The surveyed studies are categorized into three main approaches: traditional machine learning methods, deep learning techniques, and hybrid frameworks. In addition, we present benchmark datasets commonly used for the evaluation of AI-generated text detection systems. Figure 2.1 presents an overview of the main categories of AI-generated text detection methods discussed throughout this chapter.

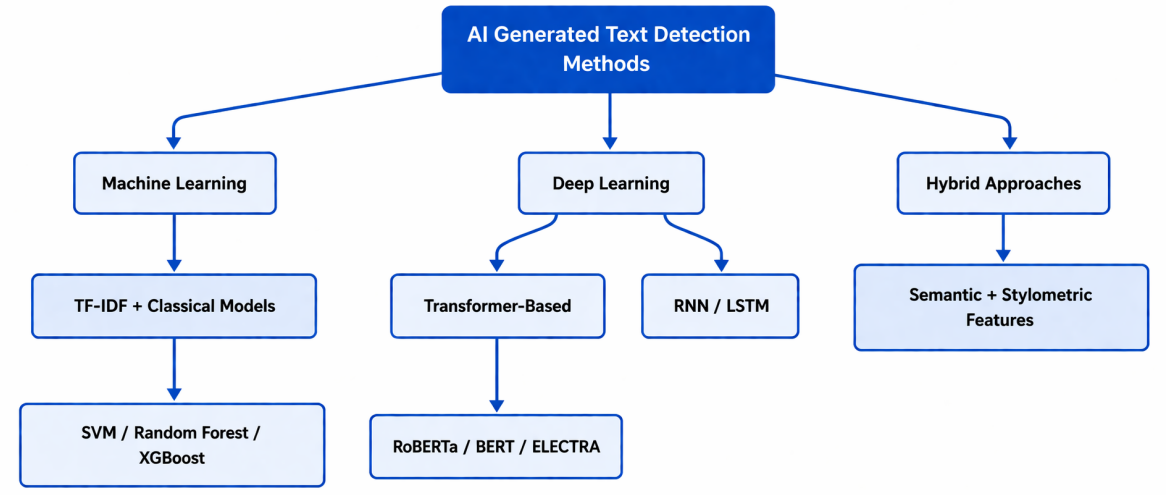


Figure 2.1: Taxonomy of AI-generated text detection methods.

2.2 Machine Learning Approaches for AI-Generated Text Detection

Several studies have investigated traditional machine learning approaches for AI-generated text detection. Islam et al. [2023] examined the distinction between human-written and ChatGPT-generated text using a dataset of 10,000 English-language samples collected from news and social media platforms, alongside outputs generated by GPT-3.5. After preprocessing and TF-IDF feature extraction, several classifiers were evaluated, where the Extremely Randomized Trees (ERT) model achieved the highest accuracy of 77%, outperforming Logistic Regression, Random Forest, Support Vector Machines, AdaBoost, Bagging, and Multi-Layer Perceptron models, whose performance ranged between 69% and 75%.

In a related study, Alamleh et al. [2023] constructed a publicly available dataset of 500 samples from computer science assignments and quizzes to evaluate classical machine learning models in essay-based and programming-oriented contexts. Using TF-IDF representations, several classifiers were evaluated, including Logistic Regression, Decision Trees, Support Vector Machines, Neural Networks, and Random Forests. Experimental results showed that Random Forest achieved the best performance, with accuracies of 93% for essay prompts, 93.5% for programming prompts, and 92.5% on the combined dataset. The study further highlighted that classical approaches outperformed deep learning models due to the limited dataset size and noted increased difficulty when handling programming-oriented content.

Prova [2024] constructed a balanced dataset of 3,000 samples and evaluated both classical machine learning models, including XGBoost and SVM, and a transformer-based approach. After applying preprocessing steps such as stopword removal, stemming, lemmatization, and feature extraction using CountVectorizer, the transformer model achieved the highest accuracy of 93%, surpassing XGBoost (84%) and SVM (81%), demonstrating its ability to capture subtle linguistic patterns relevant to this task.

To evaluate performance across diverse textual genres, Hayawi et al. [2024] introduced a dataset comprising essays, stories, poetry, and Python code generated by both humans and large language models such as GPT and BARD. The study reported strong performance in binary classification tasks, where Support Vector Machines achieved accuracies of 99.85% for Human vs. GPT and 99.66% for Human vs. BARD. In the multiclass setting, Random Forest obtained an accuracy of 95.74%. The results also showed a decline in performance for creative genres such as stories and poetry, indicating the increased difficulty of detecting more expressive writing styles. The authors further highlighted limitations including genre imbalance, and incomplete category coverage.

Zain et al. [2025] proposed a multi-strategy approach using the M-DAIGT dataset [Lamsiyah et al., 2025], which consists of 30,000 balanced samples from news articles and academic writing generated by multiple large language models, including GPT-4 and Claude. As a baseline, TF-IDF features with n-grams were classified using a Linear SVM. The approach achieved an F1-score of 97.91% for news articles, and up to a 99.85% F1-score for academic writing, demonstrating the strong performance of classical machine learning techniques on large-scale balanced datasets.

Focusing on educational content, Najjar et al. [2025] introduced the CyberHumanAI dataset, comprising 1,000 cybersecurity-related paragraphs evenly divided between human-written and ChatGPT-generated text. The study evaluated traditional machine learning models, including Random Forest, SVM, J48, and XGBoost, alongside deep learning architectures such as CNNs and DNNs. XGBoost achieved the highest accuracy of 83%, outperforming deep learning models and GPTZero. Using LIME for explainable analysis, the authors showed that human-authored text tended to contain practical and action-oriented terminology, whereas machine-generated content was generally more formal and abstract. The study further highlighted challenges associated with short text length and limited dataset size.

2.3 Deep Learning Approaches for AI-Generated Text Detection

Recent research has extensively explored deep learning methods for AI-generated text detection, demonstrating the effectiveness of transformer-based architectures in distinguishing machine-generated from human-written content.

Islam et al. [2023] investigated neural network architectures, focusing on Multi-Layer Perceptron (MLP) and BiLSTM models. While achieving moderate accuracies of approximately 75–76%, their results demonstrated the ability of neural networks to capture complex linguistic and contextual dependencies essential for distinguishing human and AI-generated text.

Chen et al. [2023] introduced the OpenGPTText dataset, comprising human-written texts paired with GPT-3.5-Turbo paraphrases, and evaluated transformer-based detectors using RoBERTa and T5. Both models achieved over 97% accuracy on the test set, demonstrating the efficacy of fine-tuned transformers in AI-generated text detection and providing a benchmark for future studies.

Yadagiri et al. [2024] employed a fine-tuned RoBERTa model on the HC3-English dataset [Guo et al., 2023], incorporating both deep contextual representations and engineered linguistic features, such as part-of-speech tags, vocabulary richness, readability metrics, perplexity, and burstiness. Among several evaluated architectures, including RNN, CNN-BiLSTM, BERT, and GPT-2, RoBERTa achieved the highest accuracy of 99.73%, highlighting the benefits of combining transformer-based embeddings with explicit linguistic cues.

Similarly, Wang et al. [2024a] investigated a BERT-based binary classification approach for AI-generated text detection. The fine-tuned model achieved a test accuracy of 97.71%, demonstrating the effectiveness of transformer-based models in distinguishing AI-generated text from human-written content.

In parallel, Mo et al. [2024] emphasized the importance of balanced datasets and hybrid deep learning architectures for robust detection. Their work introduced a near-balanced binary classification dataset consisting of 708 human-authored texts and 670 AI-generated texts. The proposed architecture integrated LSTM, CNN, and transformer-based attention modules to capture both local lexical patterns and long-range contextual dependencies. Experimental results indicated near-perfect classification accuracy; however, the study noted limitations regarding text length variability and domain transferability, highlighting the need for diverse datasets to ensure generalization.

AL-Smadi [2025] proposed a transformer-based approach using ELECTRA for detecting AI-generated English essays. The model was fine-tuned on a corpus comprising 1,467 AI-generated and 629 human-written essays from the ETS Corpus of Non-Native Written English, alongside outputs from multiple LLMs. By integrating stylometric features such as sentence length and vocabulary richness, the approach achieved an F1-score of 99.7% on the validation and test sets, underscoring the effectiveness of combining transformer representations with stylistic features.

Recent research has increasingly investigated **contrastive learning** as a powerful approach for AI-generated text detection and source attribution. Within this line of work, various methods have been proposed that exploit **contrastive learning** to obtain more effective text representations, thereby improving the ability to distinguish between different sources and enhancing overall attribution performance in AI-generated text detection tasks.

The TRACE framework proposed by Wang et al. [2024b] addresses source attribution in large language models using supervised contrastive learning, TF-IDF-based sentence selection, and a modified SBERT encoder to generate discriminative embeddings. The framework employs KNN and centroid-based classification, while UMAP is used for visualization and interpretability. Experiments conducted on datasets such as BooksSum, DBpedia-14, and CC-News achieved accuracy ranging from 80% to 96%

on the BooksSum dataset with 25 sources, while remaining effective even when the number of sources increased to 100.

The DeTeCtive framework proposed by Guo et al. [2024] investigates AI-generated text detection through writing-style representations using a multi-level, multi-task contrastive learning approach. The framework integrates encoders such as RoBERTa, SimCSE, T5, and Longformer with dense retrieval methods including KNN and cosine similarity. Evaluations on Deepfake, M4, and TuringBench datasets achieved F1-scores exceeding 92%, reaching up to 99.77% in certain scenarios. The study also demonstrated strong robustness against paraphrasing attacks and improved performance in Out-Of-Domain settings compared to existing approaches.

Recent studies on AI-generated text detection aim to improve generalization across unseen models and multilingual settings; however, methods such as [Ta et al., 2026] still face challenges under external data shifts. The proposed approach addresses this limitation using multi-level contrastive learning and multi-task learning, treating each LLM as a distinct "generator" to better capture stylistic variations. It introduces a large multilingual dataset containing 83,350 samples generated by models such as GPT, Gemini, and LLaMA, including human-written and hybrid texts. The framework achieves strong performance, reaching 95.58% accuracy on this dataset and over 96% on benchmarks such as LLM-DetectAIve and HART, demonstrating improved robustness and generalization on unseen data.

2.4 Hybrid Approaches

SenDetEX proposed by Zhang et al. [2024] is a sentence-level AI-generated text detection framework that integrates stylistic and contextual representations to improve detection robustness. It introduces the AutoFill-Refine strategy to construct high-quality human–AI hybrid texts and employs a proxy model (Mproxy) to enhance contextual feature learning. The framework is evaluated on datasets such as XSUM

and WritingPrompts using generators including DeepSeek-V3 and GPT-4o, and compared with methods like Log Probability, DNA-GPT, Fast-DetectGPT, SeqXGPT, and POGER under both in-domain and cross-domain settings. Experimental results show that SenDetEX consistently outperforms prior approaches, achieving up to 97.7% F1 in in-domain evaluations and up to 96.2% F1 in cross-domain settings, demonstrating strong robustness and generalization capability.

Zeng et al. [2024] introduced the CoAuthor dataset for studying human–AI collaborative writing, consisting of 1,445 essays written by 63 authors interacting with GPT-based assistants, and enabling sentence-level classification into human, AI-generated, and collaborative text. The authors propose a two-stage pipeline that combines sentence segmentation (e.g., Transformer2, SegFormer, TriBERT) and classification using models such as BERT and DeBERTa-v3. Experimental results show that the best configuration (Perfect Segment Detector + DeBERTa-v3) achieves a Kappa score of 0.5166, while realistic segmentation methods range between 0.30 and 0.50, and a sentence-level baseline reaches approximately 0.3447, highlighting the strong impact of segmentation quality on detection performance.

Knudsen and Hardmeier [2025] addresses AI-generated text detection to mitigate challenges such as misinformation and academic misuse. It introduces the AI Text Detection Pile dataset, containing approximately 680,000 balanced human and AI-generated samples sourced from platforms like Reddit and generated using models such as GPT-2, GPT-3, and ChatGPT. The study evaluates three approaches: a feature-based XGBoost model using 917 linguistic and statistical features, a fine-tuned DistilBERT classifier, and a hybrid model combining handcrafted features with transformer embeddings through a feed-forward neural network. Results show that DistilBERT achieves the best performance on clean data with a 98.3% F1-score, while the hybrid model reaches 94.1% F1-score and demonstrates stronger robustness under adversarial conditions, with only a 4.9% performance drop, highlighting the benefit of combining linguistic features with contextual embeddings under distribution shifts.

2.5 Datasets for AI-Generated Text Detection

In this section, we introduce the benchmark datasets commonly used for AI-generated text detection, which are summarized in Table 2.1.

TuringBench

The TuringBench dataset is designed to evaluate the detection of AI-generated text in a Turing Test scenario. It contains 200,000 English text instances, including human-written texts from news articles, social media, and online forums, and AI-generated texts produced by models such as GPT-1, GPT-2, GPT-3, GROVER, and XLNet. Texts cover diverse topics and styles, with varying lengths. Texts are labeled as human or AI-written, with AI texts tagged by their generating model for attribution purposes, enabling both binary classification (human vs. AI) and authorship attribution (identifying which AI model generated a text) [Uchendu et al., 2021].

HC3

The HC3 dataset (Human ChatGPT Comparison Corpus) is a large dataset containing 40K questions and their corresponding answers from human experts and ChatGPT about multidomains (open-domain, computer science, finance, medicine, law, and psychology) in english and chinese, with human answers collected from QA datasets and wiki text, and ChatGPT answers for the same questions. However, the dataset remains unbalanced and exhibits a bias toward English [Guo et al., 2023].

RAID

The Robust AI Detection Benchmark (RAID) is a large-scale benchmark dataset for machine-generated text detection, primarily in English, with the RAID-extra extension providing additional multilingual data. It contains over six million machine-generated texts produced by 11 language models, including GPT-4, ChatGPT, and

several open-source models. The dataset covers eight domains (e.g., abstracts, books, and news) and incorporates 11 adversarial attack types as well as four decoding strategies. About 2,000 human texts per domain were sampled, and each was paired with multiple machine-generated versions produced under different settings [Dugan et al., 2024].

MAGE

The MAGE dataset is a large-scale benchmark proposed for evaluating machine-generated text detection in realistic and diverse conditions. It contains approximately 447k English text instances, which are built from human-written texts collected across 7 distinct writing tasks, such as news writing, story generation and scientific writing, as well as the corresponding machine-generated texts produced by 27 large language models, such as ChatGPT, LLaMA and BLOOM [Li et al., 2024]. In addition, the dataset supports paraphrased versions of generated texts, allowing the evaluation of model robustness against rewriting and paraphrasing strategies. For the experiments conducted in this work, the dataset was divided into training, validation, and testing subsets in order to ensure effective model training, hyperparameter tuning, and reliable performance evaluation.

CHEAT

The CHEAT dataset is specifically designed to detect spurious academic content generated by ChatGPT, consists of 15,395 human written abstracts from IEEE Xplore in the field of computer science (including NLP, vision, machine learning) and 35,304 ChatGPT-written abstracts created by generating new abstracts, polishing human text, mixing human and generated content to simulate adversarial examples. All abstracts are in English, with an average length of 163.9 words and a total vocabulary size of 130,272 [Yu et al., 2025].

Table 2.1: Overview of Datasets Used for AI-Generated Text Detection

Dataset	**Size**	**Lang.**	**Domains**	**Models**
TuringBench [Uchendu et al., 2021]	200K	English	News, social media, forums	GPT-1, GPT-2, GPT-3, Grover, XLNet
HC3 [Guo et al., 2023]	40K	English, Chinese	Multiple domains	ChatGPT
RAID [Dugan et al., 2024]	6M	English	8 domains	11 LLMs
MAGE [Li et al., 2024]	447K	English	News, stories, scientific writing, etc.	ChatGPT, LLaMA, BLOOM, etc.
CHEAT [Yu et al., 2025]	50K	English	CS abstracts	ChatGPT

2.6 Conclusion

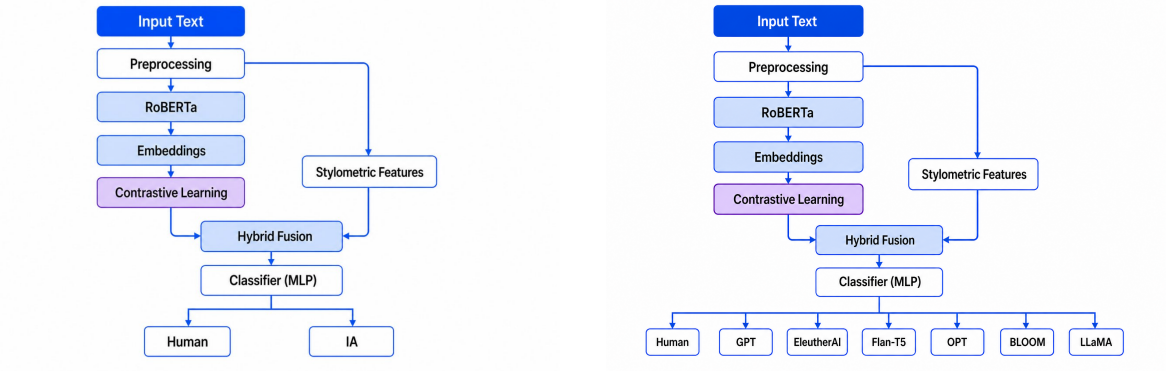
This chapter has provided a comprehensive overview of research on AI-generated text detection and source attribution, covering traditional machine learning, deep learning, and hybrid approaches. It has also highlighted commonly used benchmark datasets and their role in training and evaluating detection systems. The next chapter presents the proposed methodology, implementation details, as well as experiments and results.

Chapter 3

Methodology

3.1 Introduction

This chapter presents the proposed approach for AI-generated text detection and source attribution. The method is based on a hybrid architecture organized as a two-branch system. The semantic branch captures contextual meaning from the input text using a pre-trained RoBERTa(Robustly Optimized BERT Pretraining Approach) [Liu et al., 2019] model, selected for its strong contextual representation capabilities and proven effectiveness in text classification tasks. It is further enhanced with a supervised contrastive learning strategy to improve the structure and separability of the learned embeddings. In parallel, the stylometric branch models writing style characteristics using handcrafted linguistic and statistical features. The outputs of both branches are then fused to form a unified representation, which is used for final classification. An overview of the proposed framework is illustrated in Figure 3.1, which summarizes the main components of the model and their interactions.



(a) Binary Classification Task. (b) Multi-class Attribution Task.

Figure 3.1: Proposed framework for AI-generated text detection and source attribution tasks.

3.2 Dataset and Data Preparation

This section describes the dataset used to evaluate the proposed framework, along with the preprocessing pipeline applied to prepare the raw text samples for feature extraction and model training.

3.2.1 Dataset Overview

The experiments in this work were conducted using the MAGE dataset[Li et al., 2024], following its original training, validation, and test partitions. The dataset contains both human-written and AI-generated texts, enabling binary detection as well as multi-class attribution tasks. The overall sample distribution is presented in Table 3.1. To address class imbalance in the binary classification setting, the number of samples per class was limited to a maximum of 50,000 instances during training. For the multi-class setting, the task is formulated as an LLM-source attribution problem involving seven classes: human-written text and six LLM sources, namely **OpenAI GPT, LLaMA, OPT, Flan-T5, BLOOM, and EleutherAI**. The GLM (General Language Model) family was excluded due to insufficient representative samples. To ensure balanced training, each class was capped at 11,585 instances, corresponding to the size of the smallest class.

Table 3.1: Statistical Overview of the MAGE Dataset

Subset	**Total**	**AI**	**Human**	**AI (%)**	**Human (%)**
Train	319,071	93,318	225,753	29.25	70.75
Validation	56,792	28,799	27,993	50.71	49.29
Test	60,743	30,265	30,478	49.82	50.18
Total	**436,606**	**152,382**	**284,224**	**34.90**	**65.10**

3.2.2 Data Preprocessing Pipeline

We start by preparing the text data to make it consistent. First, we remove URLs and normalize extra whitespace while keeping the meaning of the text unchanged. The cleaned texts are then tokenized using the *RoBERTa-base* tokenizer. We apply padding and truncation to a fixed length of 512 tokens so that all inputs have the same size. Finally, the tokenized outputs are converted into input IDs and attention masks and organized into DataLoaders with a batch size of 16 for training and evaluation.

3.3 Semantic Embeddings

The semantic branch is built on a pre-trained *RoBERTa-base* encoder, which produces contextual representations for each input text. The hidden state corresponding to the [**CLS**] token from the final transformer layer is used as a global embedding of the sequence.

This representation is then used in two parallel components. The first branch applies dropout followed by a fully connected layer to produce classification logits, which are optimized using cross-entropy loss. The second branch passes the same [**CLS**] embedding through a projection head composed of linear layers, batch normalization, and Rectified Linear Unit (ReLU) activation. The output of this projection head is L2-normalized to ensure stable similarity computations in the embedding space.

3.3.1 Supervised Contrastive Learning

Within the proposed model, supervised contrastive learning is applied to improve the quality of the semantic embeddings learned by the model. Instead of relying only on classification loss, we encourage the model to learn a structured embedding space where samples from the same class are closer together.

For each mini-batch, embeddings produced by the projection head are compared using cosine similarity. Positive pairs are defined as samples belonging to the same class, while all other samples in the batch are treated as negatives.

The contrastive loss is computed jointly with the cross-entropy loss, and the final training objective is defined as:

$$\mathcal{L} = \mathcal{L}_{CE} + \alpha \mathcal{L}_{SupCon}$$

where $\alpha = 0.1$ controls the contribution of the contrastive term.

This combination allows the model to learn both discriminative decision boundaries and well-structured semantic embeddings, which improves performance on AI-generated text detection and source attribution tasks.

3.4 Stylometric Features

To complement the semantic branch, a set of 20 handcrafted stylometric features is extracted from each input sample in order to capture structural and statistical writing patterns that may differ between human-written and AI-generated text. These features are grouped into four main categories:

1. **Lexical richness:** measures vocabulary diversity and lexical complexity. It includes the Type-Token Ratio (TTR), which represents the proportion of unique

words relative to the total number of words, the Hapax Legomena Ratio, which measures the proportion of words appearing only once in the text, and the Average Word Length, which reflects the average number of characters per word. Together, these features characterize vocabulary diversity and lexical complexity.

2. **Sentence-level statistics:** describe the structural organization of the text. These features include the average sentence length, the standard deviation of sentence lengths, and burstiness, which quantifies the variability of sentence lengths throughout a document. In addition, normalized frequencies of punctuation marks, including commas, periods, and question marks, are computed to capture stylistic writing habits

3. **Syntactic composition:** analyzes the grammatical structure of the text through Part-of-Speech (POS) tagging. Specifically, the proportions of nouns, verbs, adjectives, pronouns, and function words are computed relative to the total number of words. These grammatical distributions characterize an author's writing style
.

4. **Distributional and fluency features:** describe higher-level statistical properties of the text. Word entropy and bigram entropy measure the diversity and unpredictability of words and consecutive word pairs, respectively, while the repetition score quantifies the amount of repeated content. The compression ratio estimates textual redundancy by measuring how efficiently the text can be compressed, and character-level entropy captures the variability of character usage. In addition, text perplexity is computed using a pre-trained **GPT-2** language model as a measure of textual predictability. Lower perplexity values generally indicate more predictable and machine-like text, whereas higher values are often associated with the greater variability found in human-written documents.

Finally, all extracted features are normalized using standard scaling before being fused with the semantic embeddings in the proposed hybrid framework.

3.5 Hybrid Gated Fusion

The proposed fusion module integrates semantic embeddings extracted from RoBERTa encoder, refined through supervised contrastive learning with stylometric features through an adaptive gated fusion mechanism. Both representations are first projected into a shared latent space using dedicated projection networks. The semantic embeddings are transformed through a semantic projection layer, while the stylometric features are mapped into a dense latent representation using an MLP-based projection network. The projected representations are concatenated and passed to a gating network that computes a dynamic weighting coefficient $\beta \in [0, 1]$ using a sigmoid activation function. The fused representation is defined as:

$$h_{\text{fused}} = \beta \cdot h_{\text{sem}} + (1 - \beta) \cdot h_{\text{stat}}$$

where h_{sem} and h_{stat} denote the projected semantic and stylometric embeddings, respectively. The fused embedding is then forwarded to an Multi-Layer Perceptron (MLP) classifier with nonlinear activation and dropout regularization, and the model is trained using cross-entropy loss for both binary classification and multi-class attribution, depending on the task configuration. This adaptive fusion strategy enables the model to dynamically balance semantic and stylometric information, improving the robustness and effectiveness of AI-generated text detection.

3.6 Overall Framework

The previous sections described each component of the proposed methodology independently. Figure 3.2 summarizes the complete processing pipeline of the proposed Hybrid Fusion Framework. Starting from the input text, the framework performs preprocessing before extracting semantic and stylometric representations in parallel. These representations are projected into a common latent space and combined through

an adaptive gated fusion mechanism. The fused representation is finally passed to an MLP classifier to perform either binary AI-generated text detection or multi-class source attribution.

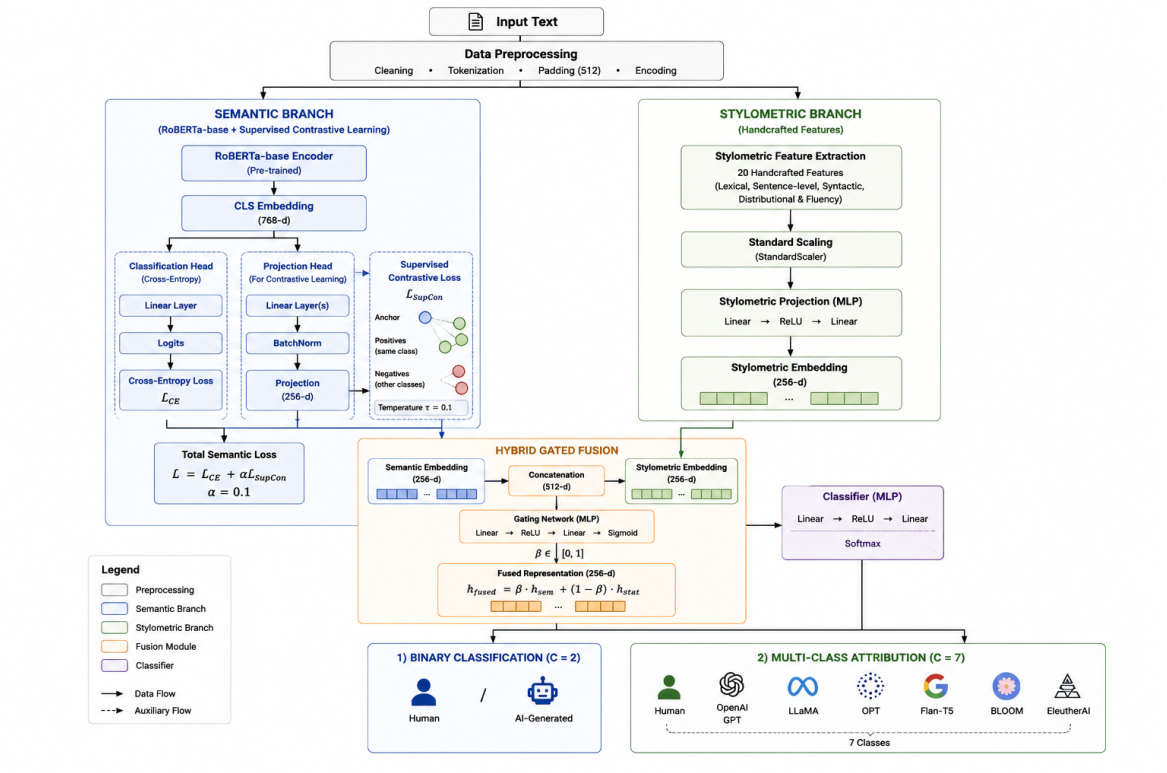


Figure 3.2: Overall architecture of Our proposed Hybrid Approach for AI-generated text detection and source attribution.

3.7 Conclusion

This chapter presented the proposed hybrid framework for AI-generated text detection and source attribution. It described the overall methodology, including dataset preparation, preprocessing, feature extraction, and the integration of semantic and stylometric representations for final classification. The next chapter presents the experimental setup and discusses the obtained results.

Chapter 4

Experiments

4.1 Introduction

This chapter presents the experimental evaluation of the proposed Hybrid Fusion Framework for AI-generated text detection and source attribution. The framework is evaluated on binary classification and multi-class attribution tasks using the MAGE benchmark dataset. The chapter describes the experimental setup, evaluation protocol, and performance analysis, followed by quantitative results and visualization-based analyses to assess the effectiveness of the proposed approach.

4.2 Experimental Setup

This section presents the experimental environment, implementation settings, and evaluation protocol adopted to assess the proposed framework. It describes the computational resources, software environment, hyperparameter configuration, and performance evaluation metrics used throughout the experiments.

4.2.1 Development Environment

The experiments conducted in this work were carried out using a set of well-established tools and computational resources, selected for their reliability and suitability for large-scale natural language processing tasks.

Kaggle

All experiments were conducted on Kaggle[1], a cloud-based platform for data science and machine learning. Kaggle provides free access to Graphics Processing Unit (GPU) accelerators through interactive Python notebooks, enabling efficient training of deep learning models without requiring local hardware infrastructure. In this work, two NVIDIA Tesla T4 GPUs, each with 14.56 GB of memory, and 31.35 GB of RAM were used to accelerate model training and inference.

Python

Python 3.12[2] was used as the primary programming language for all implementation, training, and evaluation pipelines in this work. Its extensive ecosystem of scientific and machine learning libraries makes it the standard choice for natural language processing research.

Python Libraries

The following Python libraries were used in this work to support data processing, model development, training, and evaluation.

[1] https://www.kaggle.com/
[2] https://www.python.org/

Transformers[3] is a library developed by Hugging Face for natural language processing tasks. It provides pre-trained transformer-based models and tokenization tools. In this work, it is used to load and fine-tune the RoBERTa model, as well as to load the GPT-2 model for perplexity-based feature computation.

PyTorch[4] is an open-source deep learning framework used for building and training neural networks. It supports dynamic computation graphs and automatic differentiation, making it well-suited for model implementation and optimization.

Scikit-learn[5] is a machine learning library providing tools for model evaluation, preprocessing, and classification. It is used to compute evaluation metrics, implement the MLP classifier, normalize features using `StandardScaler`, and visualize embeddings through t-SNE (t-distributed Stochastic Neighbor Embedding).

NLTK[6] is a natural language processing library providing tools for tokenization, sentence segmentation, and part-of-speech tagging, used here for stylometric feature extraction.

SciPy[7] is a scientific computing library used for entropy computation during stylometric feature extraction.

Pandas[8] and **NumPy**[9] are used for dataset manipulation, numerical operations, and feature array construction throughout the pipeline.

Matplotlib[10] and **Seaborn**[11] are used for plotting training curves and confusion matrices.

[3] https://huggingface.co/docs/transformers/index
[4] https://pytorch.org/
[5] https://scikit-learn.org/
[6] https://www.nltk.org/
[7] https://scipy.org/
[8] https://pandas.pydata.org/
[9] https://numpy.org/
[10] https://matplotlib.org/
[11] https://seaborn.pydata.org/

4.2.2 Hyperparameter Configuration

The proposed framework was trained in two sequential stages using the **AdamW** optimizer. The RoBERTa encoder was fine-tuned with a linear warmup scheduler over the first 10% of total training steps. The supervised contrastive loss was combined with cross-entropy loss, weighted by α and controlled by a temperature parameter τ to regulate embedding separation. Dropout regularization was applied in both stages to reduce overfitting, and early stopping based on validation performance was employed to prevent unnecessary training. The complete hyperparameter configuration is summarized in Table 4.1.

Table 4.1: Training Hyperparameters Used in the Proposed Framework

Hyperparameter	**Value**
Pre-trained Encoder	roberta-base
Semantic Embedding Dimension	768
Max Sequence Length	512
Batch Size (Semantic Model)	16
Batch Size (Fusion Model)	128
Learning Rate (RoBERTa Model)	2×10^{-5}
Learning Rate (Fusion Model)	1×10^{-3}
Optimizer	AdamW
Weight Decay (Semantic Model)	0.01
Weight Decay (Fusion Model)	1×10^{-4}
Number of Epochs (Semantic Model)	3
Number of Epochs (Fusion Model)	30
Warmup Ratio	0.1
Dropout Rate	0.2 – 0.3
Contrastive Loss Weight (α)	0.1
Temperature (Contrastive Loss)	0.1
Stylometric Feature Dimension	20
Projection Dimension	256
Hidden Dimension (Fusion)	256
Early Stopping Patience	7
Loss Function	Cross-Entropy + Supervised Contrastive Loss

4.2.3 Evaluation Metrics

To evaluate the performance of the proposed framework, four standard metrics were used: Accuracy, Precision, Recall, and F1-score, all derived from the confusion matrix.

Accuracy measures the proportion of correctly classified samples:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Precision measures the proportion of correctly predicted positive samples:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

Recall measures the proportion of correctly identified positive samples:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

F1-score is the harmonic mean of Precision and Recall:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

For the multi-class setting, macro-averaged metrics are computed as:

$$\text{Metric}_{\text{macro}} = \frac{1}{C} \sum_{i=1}^{C} \text{Metric}_i \tag{4.5}$$

where C denotes the number of classes. Additionally, classification reports and confusion matrices are used to provide detailed per-class evaluation and to analyze misclassification patterns.

4.3 Results and Discussion

This section presents the experimental results obtained from the proposed framework and compares them against several baseline approaches.

4.3.1 Binary Classification Results

The results reported in Table 4.2 demonstrate the effectiveness of the proposed Hybrid Framework for binary AI-generated text detection. Traditional machine learning models achieve relatively lower performance, with Macro F1-scores ranging from 55.98% to 66.91%, indicating their limited ability to capture complex semantic and stylistic patterns in the data. In contrast, the proposed Hybrid Framework significantly improves performance, reaching an accuracy of 93.20%, and outperforming both the stylometric-only and semantic-only baselines.

The improvement over the semantic-only baseline confirms that stylometric features provide complementary discriminative information beyond contextual semantic representations learned by transformer models. Furthermore, the ablation results highlight the benefit of integrating both feature types within a unified architecture.

In addition, the gate analysis indicates that semantic representations dominate the decision process for both classes, with average gate coefficients of $\beta = 0.703$ for human-written samples and $\beta = 0.838$ for AI-generated samples. This suggests that while semantic information is the primary contributor to the prediction, stylometric features still provide complementary information, particularly for human-written text. The confusion matrix obtained using the proposed framework is presented in Figure 4.1.

Table 4.2: Binary Classification Performance Comparison.

Model	Accuracy	Precision	Recall	Macro F1
Traditional Machine Learning Baselines				
Logistic Regression	0.6511	0.6526	0.6511	0.6502
Decision Tree	0.6069	0.6069	0.6069	0.6069
Random Forest	0.6695	0.6702	0.6695	0.6691
SVM	0.5600	0.5602	0.5600	0.5598
Ablation Study				
Stylometric Only	0.8416	0.8424	0.8422	0.8416
Semantic Only	0.9266	0.9267	0.9269	0.9266
Proposed				
Hybrid Framework	**0.9320**	**0.9321**	**0.9323**	**0.9320**

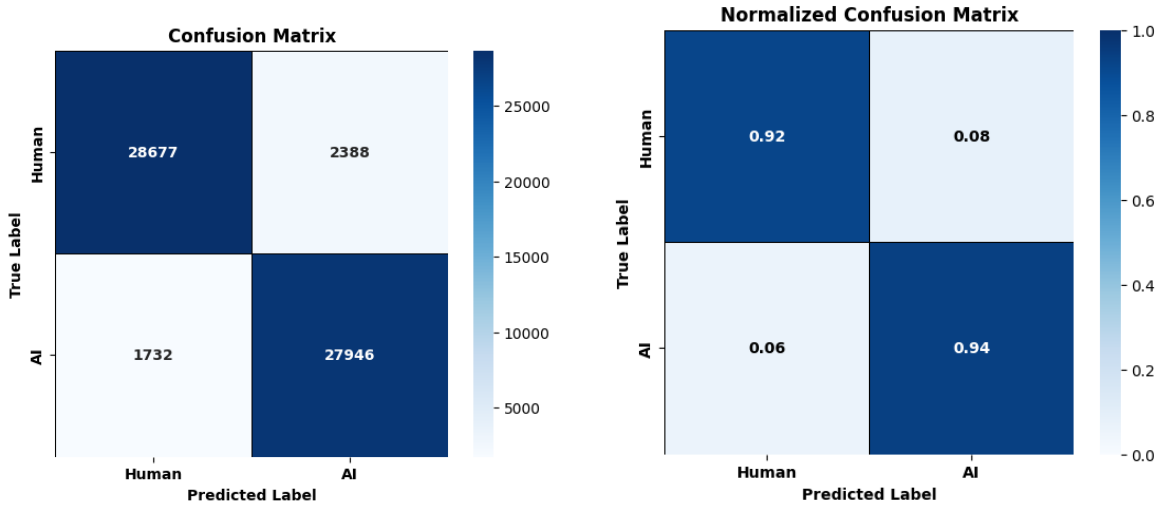


Figure 4.1: Confusion matrices for the binary classification task.

4.3.2 Multi-Class Attribution Results

The proposed framework was evaluated on a seven-class classification task comprising human-written text and six LLM-generated source categories. As reported in Table 4.3, the Hybrid Fusion Framework achieves the best overall performance, reaching a Macro F1-score of 82.22%, outperforming both traditional machine learning baselines and ablation variants.

In comparison, classical machine learning models such as Logistic Regression, SVM, and Random Forest obtain substantially lower performance, with Macro F1-scores ranging between 0.3293 and 0.4698, indicating limited capability in capturing the complex semantic and stylistic variations across LLM-generated texts. In contrast, the ablation study demonstrates that semantic representations significantly outperform stylometric features, achieving a Macro F1-score of 81.59% compared to 65.55% for stylometric-only features. This confirms the strong discriminative power of transformer-based embeddings while also highlighting the complementary role of stylometric information.

Table 4.3: Multi-Class Attribution Performance Comparison.

Model	**Accuracy**	**Precision**	**Recall**	**Macro F1**
Traditional Machine Learning Baselines				
Logistic Regression	0.4735	0.4690	0.4735	0.4698
Decision Tree	0.3453	0.3412	0.3453	0.3293
Random Forest	0.4527	0.4589	0.4527	0.4202
SVM	0.4588	0.4566	0.4588	0.4536
Ablation Study				
Stylometric Only	0.6595	0.6612	0.6595	0.6555
Semantic Only	0.8165	0.8174	0.8165	0.8159
Proposed				
Hybrid Framework	**0.8227**	**0.8222**	**0.8227**	**0.8222**

Further analysis using per-class evaluation, reported in Table 4.4, shows that LLaMA achieves the highest performance with an F1-score of 91%, followed by OpenAI GPT with 88%. Human-written text is also accurately identified with an F1-score of 84%. In contrast, lower performance is observed for OPT, FLAN-T5, and BLOOM, which can be attributed to their similar architectural designs and overlapping generative patterns.

Table 4.4: Per-Class Performance for Multi-Class Attribution

Class	Precision	Recall	F1-score
Human	0.85	0.83	0.84
OpenAI GPT	0.86	0.90	0.88
LLaMA	0.91	0.91	0.91
OPT	0.78	0.77	0.78
Flan-T5	0.77	0.79	0.78
BLOOM	0.77	0.72	0.75
EleutherAI	0.81	0.84	0.83
Macro Average	**0.82**	**0.82**	**0.82**

These findings are further supported by the confusion matrix in Figure 4.2, which reveals noticeable misclassification between OPT and EleutherAI, as well as between FLAN-T5 and BLOOM. These patterns indicate a higher degree of overlap among models with similar decoder-only architectures, making fine-grained attribution more challenging.

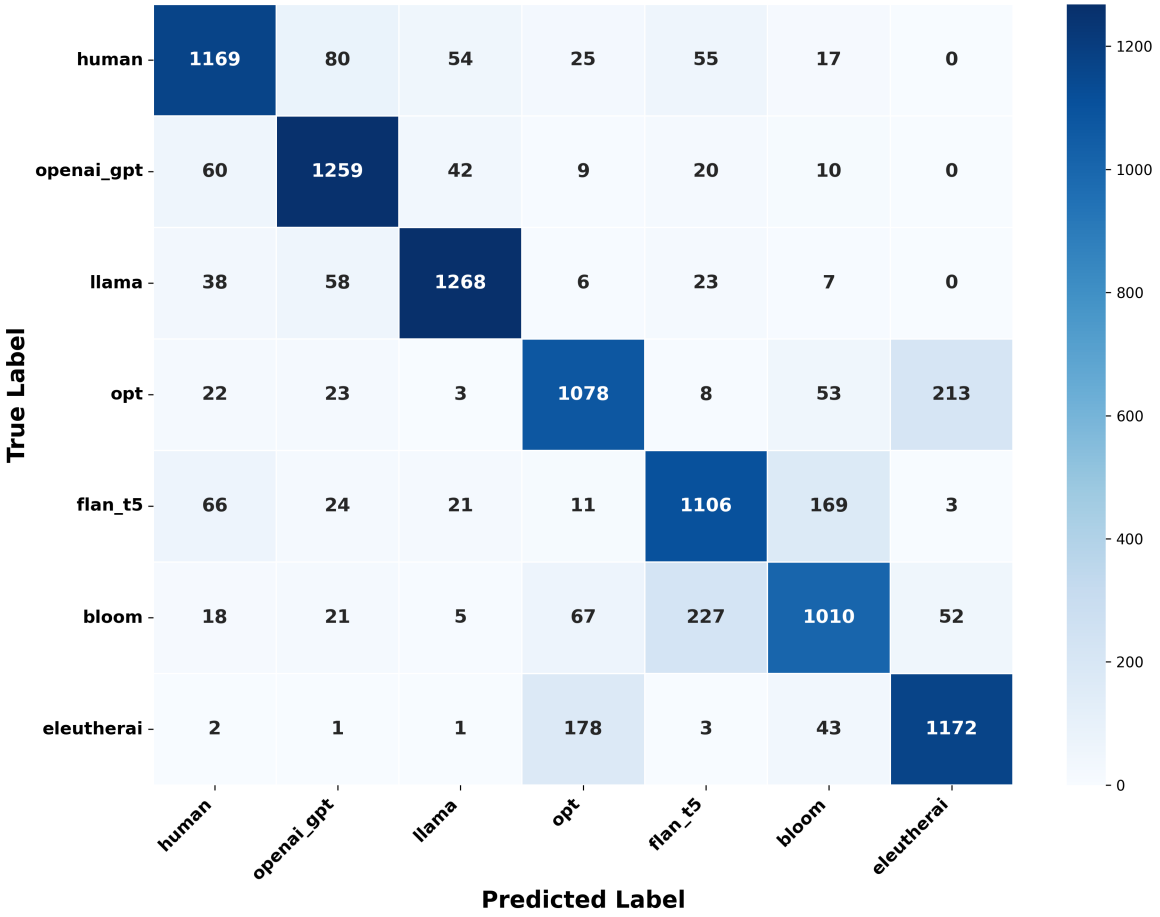


Figure 4.2: Confusion Matrix of the Multi-Class Attribution Task.

Finally, the analysis of the adaptive gating mechanism shows that the fusion model consistently assigns higher weights to semantic CLS embeddings, with gating coefficients ranging from 0.79 to 0.91 across all classes. These observations are consistent with the binary classification results, where semantic representations also dominated the decision process. This consistency across tasks confirms that semantic embeddings constitute the primary discriminative signal in the proposed framework, while stylometric features contribute as complementary information for refining class boundaries.

4.3.3 Ablation Study on Contrastive Learning

Despite integrating Supervised Contrastive Learning (SCL) jointly with cross-entropy (CE) loss, the experimental results showed no consistent improvement over the CE-only baseline in either binary or multiclass settings. Several factors reported in the literature may explain this observation.

First, Gunel et al. [2021] reported that the most significant gains from supervised contrastive learning are generally observed in few-shot and low-resource settings. In contrast, our experiments were conducted on substantially larger datasets, where conventional cross-entropy fine-tuning may already learn sufficiently discriminative representations.

Second, although a projection head was introduced to partially decouple the contrastive and classification objectives, both losses still jointly optimize the same RoBERTa encoder. As discussed by Moukafih et al. [2022], supervised contrastive fine-tuning can be viewed as a multi-objective optimization problem in which the cross-entropy and contrastive objectives may induce conflicting optimization directions. In our implementation, both losses were combined using a fixed weighting coefficient α, which may have limited the model's ability to balance the two objectives effectively.

Third, the effectiveness of supervised contrastive learning is strongly influenced by batch size. Khosla et al. [2021] showed that larger batch sizes improve contrastive representation learning by increasing the number of informative positive and negative pairs available within each batch. In our experiments, the batch size was limited to 16 due to hardware constraints, which may have reduced the effectiveness of the contrastive objective.

Taken together, these factors may explain the limited impact of supervised contrastive learning in our experiments and highlight practical challenges in applying contrastive learning to large-scale text classification tasks.

4.4 Conclusion

This chapter presented the experimental evaluation of the proposed Hybrid Fusion Framework for AI-generated text detection and source attribution. The results on both binary and multi-class tasks demonstrate that the proposed model consistently outperforms traditional machine learning baselines and ablation variants. The findings confirm that combining semantic and stylometric features through adaptive fusion improves classification performance, with semantic representations playing the dominant role while stylometric features provide complementary information.

In addition, the ablation study on supervised contrastive learning showed that its integration did not lead to consistent improvements over the cross-entropy baseline, highlighting its limited effectiveness under the current experimental conditions, and motivating further investigation into more principled optimization strategies.

Conclusion

The rapid advancement of Large Language Models (LLMs) has significantly improved the ability of artificial intelligence systems to generate fluent and coherent text that closely resembles human writing. While these models have driven substantial progress in natural language processing applications, they also raise critical concerns regarding the authenticity, transparency, and reliability of generated content. As generative models continue to evolve, distinguishing AI-generated text from human-written text has become increasingly challenging, making AI-generated text detection and source attribution an important and active research problem in natural language processing.

To address this problem, a hybrid architecture was proposed that integrates two complementary sources of information. First, deep contextual semantic representations are extracted using a pre-trained RoBERTa encoder, providing rich embeddings that capture the meaning of the input text. Second, a set of handcrafted stylometric features is introduced to model linguistic, structural, and statistical writing patterns. To enhance representation quality, supervised contrastive learning is incorporated to improve intra-class compactness and inter-class separability within the semantic embedding space. Finally, an adaptive gated fusion mechanism is designed to dynamically combine semantic and stylometric representations, enabling the model to balance both sources of information depending on their relative contribution.

The proposed framework is evaluated on the MAGE benchmark dataset under both binary classification and multi-class attribution scenarios. Experimental results

demonstrate that the hybrid model consistently outperforms traditional machine learning baselines as well as single-branch ablation models. The findings indicate that semantic embeddings derived from transformer-based models constitute the primary discriminative signal, while stylometric features provide complementary and stabilizing information that enhances overall robustness and classification performance. In the multi-class setting, the model further shows a strong capability in distinguishing between different large language model families, despite the similarity of their generated outputs.

Overall, this work confirms the effectiveness of combining semantic and stylometric information within a unified learning framework for AI-generated text detection. The proposed approach offers a balanced and adaptable solution that improves both detection accuracy and source attribution reliability.

Future research directions include extending the framework to more diverse and recent generative models, improving cross-domain and cross-lingual generalization, and integrating explainable AI techniques to enhance interpretability and trustworthiness in practical deployment scenarios.

Bibliography

Jun-gyu Kim, Sang-yeon Lee, and In-bok Lee. The development of an lstm model to predict time series missing data of air temperature inside fattening pig houses. *Agriculture*, 13(4):795, 2023. doi: 10.3390/agriculture13040795.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Tom B. Brown et al. Language models are few-shot learners, 2020.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, et al. Defending against neural fake news. *NeurIPS*, 2019. URL https://arxiv.org/abs/1905.12616.

Irene Solaiman, Miles Brundage, Jack Clark, et al. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*, 2019. URL https://arxiv.org/abs/1908.09203.

Zellig S. Harris. Distributional structure. *WORD*, 10(2–3):146–162, 1954. doi: 10.1080/00437956.1954.11659520.

Rajvardhan Patil et al. A survey of text representation and embedding techniques in nlp. *IEEE Access*, 11:36120–36146, 2023. doi: 10.1109/ACCESS.2023.3266377.

K. Jones. A statistical interpretation of term specificity in retrieval. *Journal of Documentation*, 60:493–502, 01 2004. doi: 10.1108/00220410410560573.

Lukáš Havrlant and Vladik Kreinovich. A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation). *International Journal of General Systems*, 46(1):27–36, 2017. doi: 10.1080/03081079.2017.1291635.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. URL https://arxiv.org/abs/1301.3781.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. volume 14, pages 1532–1543, 01 2014. doi: 10.3115/v1/D14-1162.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information, 2017. URL https://arxiv.org/abs/1607.04606.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Qiong Liu and Ying Wu. Supervised learning. 01 2012. doi: 10.1007/978-1-4419-1428-6_451.

Muhammad Usama, Junaid Qadir, Ahsan Raza, Hammad Arif, Kok-Lim Alvin Yau, Yehia Elkhatib, Asad Hussain, and Ala Al-Fuqaha. Unsupervised machine learning for networking: Techniques, applications and research challenges. *IEEE Access*, 7: 65579–65615, 2019. doi: 10.1109/ACCESS.2019.2916648.

Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018. URL https://api.semanticscholar.org/CorpusID:49313245.

Virginia R. de Sa. Learning classification with unlabeled data. In *Neural Information Processing Systems*, 1993. URL https://api.semanticscholar.org/CorpusID:9890353.

Xiao Liu et al. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2021. doi: 10.1109/TKDE.2021.3090866.

Raia Hadsell, Sumit Chopra, and Yann Lecun. Dimensionality reduction by learning an invariant mapping. pages 1735 – 1742, 02 2006. ISBN 0-7695-2597-0. doi: 10.1109/CVPR.2006.100.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020. URL https://arxiv.org/abs/1911.05722.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. URL https://arxiv.org/abs/2002.05709.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning, 2021. URL https://arxiv.org/abs/2004.11362.

Andrew Cucchiara. Applied logistic regression. *Technometrics*, 34:358–359, 03 2012. doi: 10.1080/00401706.1992.10485291.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. doi: 10.1007/BF00994018. URL https://doi.org/10.1007/BF00994018.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. doi: 10.1207/s15516709cog1402_1.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 11 1997. doi: 10.1162/neco.1997.9.8.1735.

Enkelejda Kasneci et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274, 2023. doi: 10.1016/j.lindif.2023.102274.

Susan Zhang et al. Opt: Open pre-trained transformer language models, 2022.

Hyung Won Chung et al. Scaling instruction-finetuned language models, 2022.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

BigScience Workshop et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022. URL https://arxiv.org/abs/2211.05100.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. LLaMA: Open and efficient foundation language models, 2023.

Gemini Team et al. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2024. URL https://arxiv.org/abs/2312.11805.

DeepSeek-AI. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024. URL https://arxiv.org/abs/2401.02954.

Anthropic. The claude 3 model family: Opus, sonnet, haiku. Technical report, Anthropic, 2024. URL https://www-cdn.anthropic.com/c6a80a657af445f40e31afac050f3bf76d3b1404.pdf.

Simone Grassini. Shaping the future of education: Exploring the potential and consequences of ai and chatgpt in educational settings. *Education Sciences*, 13(7):692, 2023. doi: 10.3390/educsci13070692.

Teo Susnjak. Chatgpt: The end of online exam integrity?, 2022. URL https://arxiv.org/abs/2212.09292.

Sameera Hajijama, Deven Juneja, and Prashant Nasa. Large language model in critical care medicine: Opportunities and challenges. *Indian Journal of Critical Care Medicine*, 28:523–525, 06 2024. doi: 10.5005/jp-journals-10071-24743.

Arun Thirunavukarasu, Darren Ting, Kabilan Elangovan, Laura Gutierrez Sinisterra, Ting Tan, and Daniel Ting. Large language models in medicine. *Nature Medicine*, 29, 07 2023. doi: 10.1038/s41591-023-02448-8.

Yunhe Feng, Sreecharan Vanam, Manasa Cherukupally, Weijian Zheng, Meikang Qiu, and Haihua Chen. Investigating code generation performance of chatgpt with crowdsourcing social data. pages 876–885, 06 2023. doi: 10.1109/COMPSAC57700.2023.00117.

Raphaël Khoury, Anderson R. Avila, Jacob Brunelle, and Baba Mamadou Camara. How secure is code generated by chatgpt?, 2023. URL https://arxiv.org/abs/2304.09655.

Kwok-Yan Lam, Victor C. W. Cheng, and Zee Kin Yeong. Applying large language models for enhancing contract drafting. In *LegalAIIA@ICAIL*, 2023. URL https://api.semanticscholar.org/CorpusID:260442567.

Jinzhe Tan, Hannes Westermann, and Karim Benyekhlef. Chatgpt as an artificial lawyer? In *AI4AJ@ICAIL*, 2023. URL https://api.semanticscholar.org/CorpusID:260442472.

Zecong Wang, Jiaxi Cheng, Chen Cui, and Chenhao Yu. Implementing bert and fine-tuned roberta to detect ai generated news by chatgpt, 2023. URL https://arxiv.org/abs/2306.07401.

Xiao Fang, Shangkun Che, Minjia Mao, Hongzhe Zhang, Ming Zhao, and Xiaohang Zhao. Bias of ai-generated content: An examination of news produced by large language models, 2024. URL https://arxiv.org/abs/2309.09825.

Shoumik Saha and Soheil Feizi. Almost ai, almost human: The challenge of detecting ai-polished writing, 2025. URL https://arxiv.org/abs/2502.15666.

Zijie Zeng, Shiqi Liu, Lele Sha, Zhuang Li, Kaixun Yang, Sannyuya Liu, Dragan Gašević, and Guanliang Chen. Detecting ai-generated sentences in human-ai collaborative hybrid texts: Challenges, strategies, and insights, 2024. URL https://arxiv.org/abs/2403.03506.

Serena Fariello, Giuseppe Fenza, Flavia Forte, Mariacristina Gallo, and Martina Marotta. Distinguishing human from machine: A review of advances and challenges in ai-generated text detection. *International Journal of Interactive Multimedia and Artificial Intelligence*, 2024. doi: 10.9781/ijimai.2024.12.002.

Ying Zhou, Ben He, and Le Sun. Humanizing machine-generated content: Evading ai-text detection through adversarial attack, 2024. URL https://arxiv.org/abs/2404.01907.

Niful Islam, Debopom Sutradhar, Humaira Noor, Jarin Tasnim Raya, Monowara Tabassum Maisha, and Dewan Md Farid. Distinguishing human generated text from chatgpt generated text using machine learning, 2023. URL https://arxiv.org/abs/2306.01761.

Hosam Alamleh, Ali Alqahtani, and AbdElRahman ElSaid. Distinguishing human-written and chatgpt-generated text using machine learning. pages 154–158, 04 2023. doi: 10.1109/SIEDS58326.2023.10137767.

Nuzhat Prova. Detecting ai generated text based on nlp and machine learning approaches, 2024. URL https://arxiv.org/abs/2404.10032.

Kadhim Hayawi, Sakib Shahriar, and Sujith Samuel Mathew. The imitation game: Detecting human and AI-generated texts in the era of ChatGPT and BARD. *Journal of Information Science*, 2024. doi: 10.1177/01655515241227531.

Ali Zain, Sareem Farooqui, and Muhammad Rafi. A multi-strategy approach for ai-generated text detection, 2025. URL https://arxiv.org/abs/2509.00623.

Salima Lamsiyah, Saad Ezzini, Abdelkader El Mahdaouy, Hamza Alami, Abdessamad Benlahbib, Samir El Amrany, Salmane Chafik, and Hicham Hammouchi. M-daigt: A shared task on multi-domain detection of ai-generated text, 2025. URL https://arxiv.org/abs/2511.11340.

Ayat A. Najjar, Huthaifa I. Ashqar, Omar A. Darwish, and Eman Hammad. Detecting ai-generated text in educational content: Leveraging machine learning and explainable ai for academic integrity, 2025. URL https://arxiv.org/abs/2501.03203.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. Gpt-sentinel: Distinguishing human and chatgpt generated content, 2023. URL https://arxiv.org/abs/2305.07969.

Annepaka Yadagiri, Lavanya Shree, Suraiya Parween, Anushka Raj, Shreya Maurya, and Partha Pakray. Detecting AI-generated text with pre-trained models using linguistic features. In *Proceedings of the 21st International Conference on Natural Language Processing (ICON)*. NLP Association of India (NLPAI), 2024. URL https://aclanthology.org/2024.icon-1.21/.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How close is chatgpt to human experts? comparison corpus, evaluation, and detection, 2023. URL https://arxiv.org/abs/2301.07597.

Hao Wang, Jianwei Li, and Zhengyu Li. Ai-generated text detection and classification based on bert deep learning algorithm, 2024a. URL https://arxiv.org/abs/2405.16422.

Yuhong Mo, Hao Qin, Yushan Dong, Ziyi Zhu, and Zhenglin Li. Large language model (llm) ai text generation detection based on transformer deep learning algorithm, 2024. URL https://arxiv.org/abs/2405.06652.

Mohammad AL-Smadi. Integrityai at genai detection task 2: Detecting machine-generated academic essays in english and arabic using electra and stylometry, 2025. URL https://arxiv.org/abs/2501.05476.

Cheng Wang, Xinyang Lu, See-Kiong Ng, and Bryan Kian Hsiang Low. Trace: Transformer-based attribution using contrastive embeddings in llms, 2024b. URL https://arxiv.org/abs/2407.04981.

Xun Guo, Shan Zhang, Yongxin He, Ting Zhang, Wanquan Feng, Haibin Huang, and Chongyang Ma. Detective: Detecting ai-generated text via multi-level contrastive learning, 2024. URL https://arxiv.org/abs/2410.20964.

Minh Ngoc Ta et al. FAID: Fine-grained AI-generated text detection using multi-task auxiliary and multi-level contrastive learning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3275–3296, 2026. doi: 10.18653/v1/2026.eacl-long.151.

Ye Zhang et al. Enhancing text authenticity: A novel hybrid approach for ai-generated text detection. In *2024 IEEE 4th International Conference on Electronic Technology, Communication and Information (ICETCI)*, pages 433–438, 2024. doi: 10.1109/ICETCI61221.2024.10594194.

Kasper Thomas Gartside Knudsen and Christian Hardmeier. Hybrid feature-embedding models for robust ai text detection. In *Proceedings of the 21st Conference on Natural Language Processing (KONVENS)*, pages 318–325, Germany, 2025. KONVENS. URL https://aclanthology.org/2025.konvens-1.27/. IT-University of Copenhagen, Denmark.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2001–2016. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-emnlp.172. URL https://aclanthology.org/2021.findings-emnlp.172/.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. RAID: A shared benchmark for

robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.674. URL https://aclanthology.org/2024.acl-long.674/.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. Mage: Machine-generated text detection in the wild, 2024. URL https://arxiv.org/abs/2305.13242.

Peipeng Yu, Jiahan Chen, Xuan Feng, and Zhihua Xia. Cheat: A large-scale dataset for detecting chatgpt-written abstracts. *IEEE Transactions on Big Data*, 11(3):898–906, 2025. doi: 10.1109/TBDATA.2025.3536929.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. Supervised contrastive learning for pre-trained language model fine-tuning, 2021. URL https://arxiv.org/abs/2011.01403.

Youness Moukafih, Mounir Ghogho, and Kamel Smaili. Supervised contrastive learning as multi-objective optimization for fine-tuning large pre-trained language models, 2022. URL https://arxiv.org/abs/2209.14161.

الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research

Faculty of Science and Technology

Department of Mathematics and Computer Science

جامعة غرداية

Université de Ghardaia

كلية العلوم و التكنولوجيا

قسم الرياضيات و الاعلام الالي

Ghardaia le 30/06/2026

Rapport de correction de mémoire de Master SIEC

Je soussigné M~~/Mme/Mlle~~ : **Slimane Oulad-Naoui**

Président du jury du mémoire de Master intitulé :

Hybrid Learning for AI-Generated Text Detection and Attribution

Après les corrections apportées au rapport, je déclare que les étudiants :

Asma Bensania & Malak Dahma

Sont autorisés à déposer leur manuscrit au niveau du département.

Fait et délivré pour servir et valoir ce que de droit

رئيس قسم الرياضيات و الإعلام الآلي
الحاج موسى

جامعة غرداية
قسم الرياضيات و الاعلام الآلي
كلية العلوم والتكنولوجيا

Signature

Slimane Oulad-Naoui