



BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INTELLIGENT SYSTEMS

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

RECOGNIZING A PERSON USING STYLOMETRY

ROZPOZNÁNÍ OSOBY S VYUŽITÍM STYLOMETRIE

MASTER'S THESIS

DIPLOMOVÁ PRÁCE

AUTHOR

AUTOR PRÁCE

Bc. SAMUEL ŠIMÚN

SUPERVISOR

VEDOUCÍ PRÁCE

Ing. TOMÁŠ GOLDMANN, Ph.D.

BRNO 2026

Master's Thesis Assignment



171987

Institut: Department of Intelligent Systems (DITS)
Student: **Šimún Samuel, Bc.**
Programme: Information Technology and Artificial Intelligence
Specialization: Machine Learning
Title: **Person Recognition Based on Stylometry**
Category: Artificial Intelligence
Academic year: 2025/26

Assignment:

1. Familiarize yourself with the fundamentals of stylometry, its methods, and its applications in the field of authorship recognition.
2. Study existing approaches used in stylometric text analysis, including statistical methods and modern techniques based on machine learning.
3. Propose a method for verifying text authorship through stylometry using large language models. Focus on exploring the possibilities of their application in extracting stylistic features, comparing linguistic patterns, and identifying authorial traits in texts of varying lengths and genres.
4. Implement the proposed solution in the Python programming language and develop a simple console application demonstrating its functionality.
5. Conduct experiments aimed at evaluating the accuracy of author recognition using the implemented solution and analyze the obtained results.

Literature:

- OPARA, Chidimma. StyloAI: Distinguishing AI-generated content with stylometric analysis. In: International conference on artificial intelligence in education. Cham: Springer Nature Switzerland, 2024. p. 105-114.
- HU, Xinyu, et al. TDRLM: Stylometric learning for authorship verification by Topic-Debiasing. Expert Systems with Applications, 2023, 233: 120745.
- SABOL, Radoslav; HORÁK, Aleš. Manipulative Style Recognition of Czech News Texts using Stylometric Text Analysis. RASLAN 2022 Recent Advances in Slavonic Natural Language Processing, 2022, 191.

Requirements for the semestral defence:

Fulfillment of points 1 and 2, particularly 3.

Detailed formal requirements can be found at <https://www.fit.vut.cz/study/theses/>

Supervisor: **Goldmann Tomáš, Ing., Ph.D.**
Head of Department: Kočí Radek, Ing., Ph.D.
Beginning of work: 1.11.2025
Submission deadline: 20.5.2026
Approval date: 3.11.2025

Abstract

This thesis investigates stylometry as a cognitive biometric method for authorship verification, where the goal is to determine whether two texts were written by the same author. To reduce the influence of topic similarity, a semantic-agnostic dataset construction strategy is proposed, creating positive pairs from semantically distant texts by the same author and negative pairs from semantically similar texts by different authors. The work evaluates two transformer-based approaches: encoder-based BERT/RoBERTa models with co-attention and style regularisation, and decoder-based Qwen3 models fine-tuned using LoRA. Experiments on a custom dataset and the PAN21 benchmark show that the decoder-based Qwen3-4B model trained with focal loss achieves the best performance. The results demonstrate the potential of transformer-based models for robust stylometric authorship verification.

Abstrakt

Táto diplomová práca skúma stylometriu ako kognitívnu biometrickú metódu na verifikáciu autorstva, ktorej cieľom je určiť, či boli dva texty napísané tým istým autorom. Na zníženie vplyvu tematickej podobnosti je navrhnutá stratégia tvorby sémanticky agnostickkej dátovej množiny, v ktorej sú pozitívne páry vytvorené zo sémanticky odlišných textov toho istého autora a negatívne páry zo sémanticky podobných textov rôznych autorov. Práca hodnotí dva prístupy založené na architektúre Transformer: enkóderové modely BERT/RoBERTa s mechanizmom ko-pozornosti a štýlovou regularizáciou a dekóderové modely Qwen3 doladované pomocou LoRA adaptérov. Experimenty na vlastnej dátovej množine a benchmarku PAN21 ukazujú, že najlepšie výsledky dosahuje dekóderový model Qwen3-4B trénovaný s funkciou fokálnej straty. Výsledky poukazujú na potenciál transformerových modelov pri robustnej stylometrickej verifikácii autorstva.

Keywords

Stylometry, Cognitive Biometrics, Authorship Verification, Machine Learning, Deep Learning, Transformers, BERT, RoBERTa, Qwen3, LoRA, Co-Attention, Focal Loss

Klíčové slová

Stylometria, kognitívna biometria, verifikácia autorstva, strojové učenie, hlboké učenie, transformery, BERT, RoBERTa, Qwen3, LoRA, Co-Attention, Focal Loss

Reference

ŠIMÚN, Samuel. *Recognizing a person using stylometry*. Brno, 2026. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Tomáš Goldmann, Ph.D.

Rozšírený abstrakt

Táto diplomová práca sa zaoberá rozpoznávaním osoby na základe stylometrie, pričom sa zameriava na úlohu verifikácie autorstva. Cieľom tejto úlohy nie je priradiť text ku konkrétnemu autorovi z uzavretej množiny kandidátov, ale rozhodnúť, či boli dva vstupné texty napísané tou istou osobou. Takáto formulácia problému je prakticky významná najmä v oblastiach digitálnej bezpečnosti, forenznnej analýzy, online komunikácie a overovania identity v prostrediach, kde nie je dostupná úplná množina možných autorov. Stylometria je v práci chápaná ako forma kognitívnej biometrie, ktorá neidentifikuje človeka podľa fyzických vlastností, ale podľa spôsobu, akým spracúva, štruktúruje a produkuje jazykový prejav.

Východiskom práce je predpoklad, že autori si v priebehu času vytvárajú relatívne stabilné jazykové návyky, ktoré sa prejavujú v lexikálnych, syntaktických, štylistických a čiastočne aj sémantických charakteristikách textu. Tieto znaky môžu zahŕňať voľbu slov, používanie funkčných slov, dĺžku viet, interpunkciu, syntaktické štruktúry alebo celkový spôsob formulovania myšlienok. Súčasné prístupy založené na hlbokom učení, najmä transformerové architektúry, však často zachytávajú nielen štýl autora, ale aj tematickú podobnosť textov. To predstavuje zásadný problém pri verifikácii autorstva, pretože model môže rozhodnutie založiť skôr na podobnosti témy než na skutočných autorských znakoch. Tento jav je obzvlášť problematický v prípadoch, keď dvaja rôzni autori píše o podobnej téme alebo keď ten istý autor píše texty s výrazne odlišným obsahom.

Na riešenie tohto problému práca navrhuje stratégiu tvorby sémanticky agnostickej dátovej množiny. Jej cieľom je obmedziť vplyv tematickej podobnosti a podporiť učenie znakov, ktoré lepšie reprezentujú autorský štýl. Pozitívne páry sú preto vytvárané z textov toho istého autora, ktoré sú si sémanticky vzdialené. Naopak, negatívne páry sú vytvárané z textov rôznych autorov, ktoré sú si sémanticky blízke. Takto zostavené páry predstavujú náročnejší tréningový a testovací scenár, pretože model sa nemôže jednoducho spoliehať na podobnosť obsahu. Pri tvorbe dátovej množiny boli využité existujúce textové zdroje, pričom na výber párov bola použitá embeddingová reprezentácia textov a vyhľadávanie sémanticky blízkych alebo vzdialených dokumentov. Výsledkom je vlastná dátová množina určená na tréning a hodnotenie modelov v podmienkach, ktoré lepšie testujú schopnosť zachytiť štýlové vlastnosti autora.

Práca hodnotí dva hlavné prístupy založené na architektúre Transformer. Prvý prístup využíva enkóderové modely typu BERT a RoBERTa. Navrhnutá architektúra spracúva dvojicu textov pomocou zdieľaných enkóderových váh a je rozšírená o mechanizmus co-attention, ktorý umožňuje výmenu informácií medzi reprezentáciami oboch textov. Tento mechanizmus podporuje modelovanie vzájomných vzťahov medzi textami a umožňuje vytvárať vhodnejšie reprezentácie pre binárnu klasifikáciu. Súčasťou architektúry je aj vetva štýlovej regularizácie, ktorej cieľom je obmedziť nadmerné spoliehanie sa modelu na sémantickú informáciu a podporiť extrakciu štylistických znakov. Modely boli tréňované a porovnávané s použitím viacerých stratových funkcií, vrátane krížovej entropie, fokálnej straty a supervised contrastive loss.

Druhý prístup skúma použitie dekóderových jazykových modelov Qwen3 vo verifikačnej úlohe. V tomto prípade sú oba texty spojené do jednej štruktúrovanej vstupnej sekvencie a spracované ako jeden prompt. Model teda funguje ako cross-encoder, ktorý umožňuje tokenom jedného textu priamo interagovať s kontextom druhého textu. Na efektívne doladovanie väčších jazykových modelov boli použité LoRA adaptéry, ktoré umožňujú tréňovať iba malý počet dodatočných parametrov bez potreby plného pretrénovania celého modelu. Experimenty boli realizované s modelmi Qwen3-1.7B a Qwen3-4B, pričom boli porovnané konfigurácie tréňované pomocou krížovej entropie a fokálnej straty.

Experimentálne výsledky ukazujú, že oba navrhnuté prístupy sú schopné dosiahnuť vysokú úspešnosť pri verifikácii autorstva. Enkóderové modely dosiahli veľmi dobré výsledky najmä vzhľadom na svoju menšiu veľkosť a nižšie výpočtové nároky. Najlepší enkóderový model založený na RoBERTa-large dosiahol približne 0.98 ROC-AUC na vlastnej testovacej množine a približne 0.93 ROC-AUC na benchmarku PAN21. Tento výsledok ukazuje, že enkóderové architektúry zostávajú praktickou alternatívou v situáciách, kde je dôležitá efektivita inferencie, pamäťová náročnosť alebo nasadenie v prostrediach s obmedzenými hardvérovými zdrojmi.

Najlepšie výsledky však dosiahol dekóderový model Qwen3-4B trénovaný pomocou fokálnej straty. Tento model dosiahol hodnotu 0.9903 ROC-AUC na vlastnej testovacej množine a 0.9462 ROC-AUC na benchmarku PAN21. Výsledky zároveň ukazujú, že model generalizuje aj na PAN21, hoci PAN21 nebol použitý pri trénovaní, ale slúžil ako nezávislý out-of-domain benchmark. To naznačuje, že navrhnutá stratégia tvorby sémanticky agnostických párov podporuje robustnejšie učenie štýlových znakov a znižuje závislosť modelu od tematickej podobnosti. V porovnaní s existujúcimi systémami trénovanými priamo na PAN21 dosiahol navrhnutý model konkurencieschopnú výkonnosť, čo potvrdzuje potenciál dekóderových jazykových modelov pre úlohu stylometrickej verifikácie autorstva.

Súčasťou práce je aj praktická aplikácia, ktorá integruje navrhnuté modely a umožňuje ich použitie prostredníctvom príkazového rozhrania aj API. Táto implementácia demonštruje praktickú použiteľnosť navrhnutých prístupov a vytvára základ pre ďalšie experimenty alebo nasadenie v širších scenároch.

Hlavným prínosom práce je návrh komplexného rámca pre verifikáciu autorstva založeného na stylometrii a moderných transformerových modeloch. Práca prispieva najmä vytvorením sémanticky agnostickkej dátovej množiny, návrhom enkóderovej architektúry s ko-pozornosťou a štýlovou regularizáciou a experimentálnym overením dekóderových modelov doladovaných pomocou LoRA adaptérov. Dosiahnuté výsledky ukazujú, že dekóderové transformerové modely predstavujú perspektívny a stále relatívne málo preskúmaný smer v oblasti verifikácie autorstva. Zároveň však práca poukazuje na potrebu ďalšieho výskumu v oblastiach kalibrácie modelov, doménovej adaptácie, analýzy vplyvu zloženia dátovej množiny a porovnania jednotlivých architektúr z hľadiska presnosti, rýchlosti a praktických požiadaviek na nasadenie.

Recognizing a person using stylometry

Declaration

I hereby declare that this Master's thesis was prepared as an original work by the author under the supervision of Ing. Tomáš Goldmann Ph.D. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

I declare that generative AI tools were employed during the writing and development of this thesis to assist with language correction, text refinement, and code debugging. All final content, technical implementations, and academic conclusions were independently verified and decided by me, and I accept full responsibility for the entirety of this work.

.....
Samuel Šimún
May 18, 2026

Acknowledgements

I would like to express my sincere gratitude to Ing. Tomáš Goldmann Ph.D. for supervising my master's thesis and for providing valuable professional guidance and consultations throughout the preparation of this work.

Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

Contents

1	Introduction	5
2	Biometry and Stylometry	7
2.1	Biometry	7
2.2	Biometric Systems	8
2.3	Stylometry	11
2.4	Existing approaches to Stylometry Analysis	12
3	Authorship verification using Stylometry and Machine Learning	19
3.1	Machine Learning	19
3.2	Recognition based on SVM	30
3.3	Recognition based on LSTM	30
3.4	Recognition based on Transformers	31
4	Data Preparation and Establishing a Preliminary Experiment	33
4.1	Data collection	33
4.2	Data preparation	35
4.3	Preliminary Experiment Setup	40
5	Encoder-Based Authorship Verification	45
5.1	Motivation	45
5.2	Model Architecture	47
5.3	Training	51
5.4	Evaluation	57
5.5	Experiment Conclusion	59
6	Decoder-Based Authorship Verification with LoRA Fine-Tuning	60
6.1	Model Architecture	60
6.2	Training	62
6.3	Evaluation	64
6.4	Experiment Conclusion	69
6.5	Application	69
7	Conclusion	72
	Bibliography	74
A	Encoder model architecture	79

B	Dataset overview: Pair example	81
B.1	Positive example	81
B.2	Negative example	82
C	Machines used for training	85
C.1	Machine 1	85
C.2	Machine 2	85
C.3	Machine 3	85

List of Figures

2.1	Visualisation of Genuine-Impostor distributions. Image taken from [8]. . . .	10
2.2	Performance of stylometric methods across scenarios. Data taken from [29].	15
2.3	PCA plot, first 2 dimensions, contrasting 10,000 lemmatised word samples from three oeuvres. Image taken from [22].	17
3.1	Visualisation of data separation using SVM. Image taken from [28].	21
3.2	Difference between RNN Unit and LSTM unit. Image taken from [54]. . . .	23
3.3	Transformer architecture divided into encoder and decoder. Image taken from [33].	25
3.4	BERT masked-language model with token and positional embeddings, stacked transformer encoder layers, output representations, and a GELU-normalised classification head for predicting masked tokens. Image taken from [25]. . .	28
4.1	Data processing pipeline describing the pair-mining procedure.	36
4.2	Hard negative mining in the SentenceTransformer embedding space (all-MiniLM-L6-v2, $\mathbb{R}^{384}$, shown as 2-D PCA projection).	39
5.1	Encoder part of the proposed model based on BERT backbone, for full size please see the Appendix A.1	46
5.2	Decoder and classification parts of the proposed model, for full size please see the Appendix A.1	47
5.3	Visualisation of the proposed architecture using a BERT or RoBERTa encoder.	49
5.4	Training process of co-attention BERT model trained on 3 different loss functions: Cross-entropy, Focal loss and Supervised Contrastive Loss. . . .	54
5.5	Sigmoid gate activation at co-attention BERT layer 6, which is initialised with a bias of -1.5 (corresponding $\sigma(-1.5) \approx 0.182$, indicated by the dashed reference line).	54
5.6	Training process of Co-Attention RoBERTa model trained on 3 different loss functions: Cross-entropy, Focal loss and Supervised Contrastive Loss. . . .	56
5.7	Sigmoid gate activation at co-attention RoBERTa layer 8, which is initialised with a bias of -1.5 (corresponding $\sigma(-1.5) \approx 0.182$, indicated by the dashed reference line).	56
5.8	Confusion matrices of the Co-Attention model at the best- F_1 threshold on the custom test set and the PAN21 held-out benchmark. Cell colours reflect row-normalised recall per actual class; raw counts and row percentages are shown in each cell.	58

6.1	Architecture of the decoder classifier. Both texts are concatenated into a single prompted sequence and processed in one forward pass through the Qwen3 backbone. LoRA adapters are applied to all attention projection matrices. The hidden state of the final token is extracted, passed through dropout, and projected to a binary classification output.	63
6.2	Training loss curves for the decoder-based model across four configurations (cross-entropy vs. focal loss; 1.7B vs. 4B backbone).	65
6.3	Validation ROC-AUC curves for the decoder-based model across four configurations: cross-entropy loss with 1.7B and 4B parameter models (CE 1.7B, CE 4B), and focal loss with the same model sizes (Focal 1.7B, Focal 4B). .	66
6.4	Confusion matrices of the best model (QWEN3-4B, Focal loss) at the best- F_1 threshold τ^* on the custom test set and the PAN21 held-out benchmark. .	68
A.1	Full model architecture of the Encoder Decoder model for BERT backbone.	80

Chapter 1

Introduction

In an increasingly interconnected digital environment, the accurate verification of individual identity has become an important requirement in areas such as digital security, forensic analysis, and online communication. Traditional identity verification methods often rely on explicit credentials or physiological biometrics, such as fingerprints, facial geometry, or iris patterns. Although these methods are effective in many scenarios, they typically require physical presence, specialised sensors, or direct user cooperation. These limitations reduce their applicability in remote settings where identity must be inferred from digital traces.

For this reason, behavioural and cognitive biometrics have gained increasing attention. Unlike physiological biometrics, which identify individuals based on relatively static physical characteristics, cognitive biometrics focus on behavioural patterns that reflect how individuals act, communicate, or process information. Within this area, stylometry represents an important approach for analysing identity through written language. By measuring linguistic characteristics such as lexical preferences, syntactic structures, punctuation usage, and other stylistic patterns, stylometry can model an author’s writing style as a distinctive linguistic footprint.

This thesis focuses specifically on authorship verification. In this task, the goal is not to identify an author from a closed set of known candidates but to decide whether two given texts were written by the same person. This formulation is particularly relevant in forensic and digital environments, where the set of possible authors is often unknown, incomplete, or too large to model directly. Authorship verification, therefore, provides a practical framework for analysing identity in text-based communication.

Recent advances in natural language processing have led to the widespread use of transformer-based models in stylometric tasks. Encoder-based architectures, such as BERT and RoBERTa, have been commonly applied to authorship verification and related problems. However, one of the main challenges remains in separating stylistic information from semantic content. If two texts discuss similar topics, a model may rely on semantic similarity rather than genuine stylistic evidence. Conversely, two texts written by the same author may be difficult to match if they discuss different topics. This issue is especially important in cross-domain evaluation, where robustness to topic and genre variation is required.

To address this limitation, this thesis proposes a semantic-agnostic data preparation strategy and evaluates two transformer-based approaches for authorship verification. The dataset construction process creates challenging positive and negative pairs: positive pairs consist of semantically distant texts written by the same author, while negative pairs consist of semantically similar texts written by different authors. This design aims to force models to rely more on lexical and syntactic style rather than on topic similarity. On the modelling

side, the thesis investigates an encoder-based architecture enhanced with co-attention and style regularisation, as well as decoder-based models fine-tuned using LoRA adapters.

The main contributions of this thesis are the construction of a semantic-agnostic authorship verification dataset, the design and evaluation of an encoder-based verification architecture with co-attention and style regularisation, and the investigation of decoder-based transformer models for authorship verification. The proposed approaches are evaluated on both the custom dataset and the PAN21 benchmark in order to assess their performance and cross-domain robustness.

Chapter 2

Biometry and Stylometry

This chapter establishes the theoretical and practical framework of Stylometry in the broader context of biometrics. While classical biometry focuses on physiological traits (*e.g.*, *face*, *fingerprints*), stylometric features are considered cognitive traits in biometrics [38]. In this context, the person is identified not by physical characteristics, but by how they process and produce language.

2.1 Biometry

Biometry is a technical science that analyses and measures unique physical and behavioural (cognitive) characteristics of individuals to verify or identify them [36]. Unlike passwords or other types of credentials, biometric traits cannot be lost; they evolve with the person [38]. They are also considered sufficiently unique to perform tasks such as identification among the human population.

Although biometry has traditionally denoted the broader use of mathematical models in medicine and biology, in computer science it refers to the automated identification or verification of individuals based on quantifiable physiological or behavioural characteristics [51, 36]. Stylometry, long associated with forensic linguistics and authorship attribution, is increasingly being incorporated into biometric systems as a higher-level behavioural modality.

Any human psychological or cognitive trait can serve as a biometric characteristic as long as it meets the following criteria [18, 10]:

- **Universality** - Each person should have their own characteristic that is shared by the population.
- **Distinctiveness** - Any two persons should have differences in terms of biometric characteristics.
- **Permanence** - A person's biometric traits should remain stable or unchanged for a long period of time.
- **Collectability** - The characteristic should be quantifiable and collectable.
- **Performance** - Refers to the accuracy, speed, and robustness of the biometric system.
- **Acceptability** - The extent to which people are willing to accept the use of a particular biometric.

- **Circumvention** - How easily can the system be fooled by fraudulent methods or artefacts.

2.1.1 Physiological biometry

Physiological biometry relies on the analysis of static anatomical structures of the human body. Recognition based on physiological traits is generally innate and exhibits high levels of performance [19]. These characteristics make them very popular for deployed forms of biometric security. While various physiological characteristics can be used for human identification, including iris patterns and vascular structures in the hand, fingerprint and facial recognition remain the most widely adopted modalities.

Fingerprint recognition [19] is the most established biometric modality. A fingerprint consists of several patterns of ridges and valleys on its surface. These ridges and valleys are formed in the womb and remain largely unchanged throughout a person's life. The uniqueness of fingerprints is based on local ridge characteristics called minutiae [18]. Two primary minutiae types are *ridge endings*, which denote points where a ridge abruptly terminates, and *ridge bifurcations*, which denote points where a single ridge forks into two branch ridges [19].

Facial recognition is a common, non-intrusive¹ biometric method that identifies individuals by analysing either specific facial features like eyes and nose, or global patterns like canonical face combinations [19]. This type of recognition is sensitive to environmental conditions [18]. The next problem could be angle variance from which the sample of the face was taken, or reliability, which is a debated issue.

2.1.2 Behavioural biometry

Behavioural biometry is part of biometry, which analyses how a person conducts an action or interacts with a digital platform [36]. Unlike physical traits, which are generally checked at a single point of entry, behavioural biometrics can be monitored passively and continuously throughout a session to detect deviations that might indicate a person's identity.

A typical example of behavioural biometry is Voice Recognition [36], which combines biological and behavioural characteristics. The sound of an individual depends on multiple factors, including biological aspects of the human body, such as the mouth, nose, and lips, as well as intonation and other behavioural traits.

2.2 Biometric Systems

We can define biometric systems [36] as pattern recognition systems that recognise a person based on a feature vector derived from physiological or behavioural traits provided by the person.

There are two distinct modes in which a biometric system can operate, depending on the identity claim made by the user [36, 18]. The first one is *Verification (1:1 matching)*. In this mode, we address the question, “*Am I who I claim to be?*”. The user provides their identity, and the system performs a one-to-one comparison between the live sample and the sample associated with that identity. This mode is often used in commercial applications, such as unlocking a smartphone with a fingerprint or an iris scan. The second mode used

¹Non-intrusive biometrics stand for identification without requiring physical contact or active cooperation.

is *Identification (1:N matching)*. This mode addresses the question, “*Who am I?*”. The system attempts to identify or locate the user in its database using the user’s sample. This one-to-many process is commonly used in forensic investigations or border control. Additionally, the complexity and risk of false matches increase with the growth in database size (N) [18].

2.2.1 Error rate

No biometric system operates at 100%, as the performance of individual systems depends on multiple factors, such as different positioning on the acquiring sensor or imperfect imaging conditions [8]. For that reason, the output of biometric systems is typically a matching score (ranging from 0 to 1) that quantifies the similarity between two samples. This matching score provides the user with information on how certain the system is that two samples belong to the same user. This score is then compared with the system threshold to determine whether the system will *accept* or *reject* the user.

For the verification of biometric systems, negative pairs (pairs of samples that don’t belong to the same person) and positive pairs (pairs of samples that belong to the same person) are created. Based on the response matching scores from the system and known labels, we are able to determine the following measurements [8]:

- **FNMR** (*False NonMatch Rate*) - rejecting two samples that belong to the same person. The formula for this rate is [8]:

$$\text{FNMR} = \frac{\text{Number of false nonmatches}}{\text{Number of pairs}}. \quad (2.1)$$

- **FMR** (*False Match Rate*) - accepting two samples that belong to different persons. The formula for this rate is [8]:

$$\text{FMR} = \frac{\text{Number of false matches}}{\text{Number of pairs}}. \quad (2.2)$$

FNMR and FMR [8] are functions dependent on the parameter t (*threshold*). While we reduce the threshold, the system will be more tolerant, increasing the chance that it will accept two samples; however, the FMR will also increase. On the other hand, if we increase the threshold, the system will be stricter, and the chance of accepting two samples will be lower, resulting in a higher FNMR. With the dynamic threshold, we can create *genuine distributions* from positive pairs and *Impostor distributions* from negative pairs, based on responses from biometric systems. The visualisation of these dependencies can be seen in Figure 2.1.

2.2.2 Receiver operational curve (ROC)

The Receiver Operating Characteristic (ROC) [5] curve is an evaluation technique originally developed during the Second World War. For each threshold, a confusion matrix is computed, from which two rates are derived: the True Positive Rate (TPR, also called sensitivity or recall) [5]:

$$\text{TPR} = \frac{TP}{TP + FN} \quad (2.3)$$

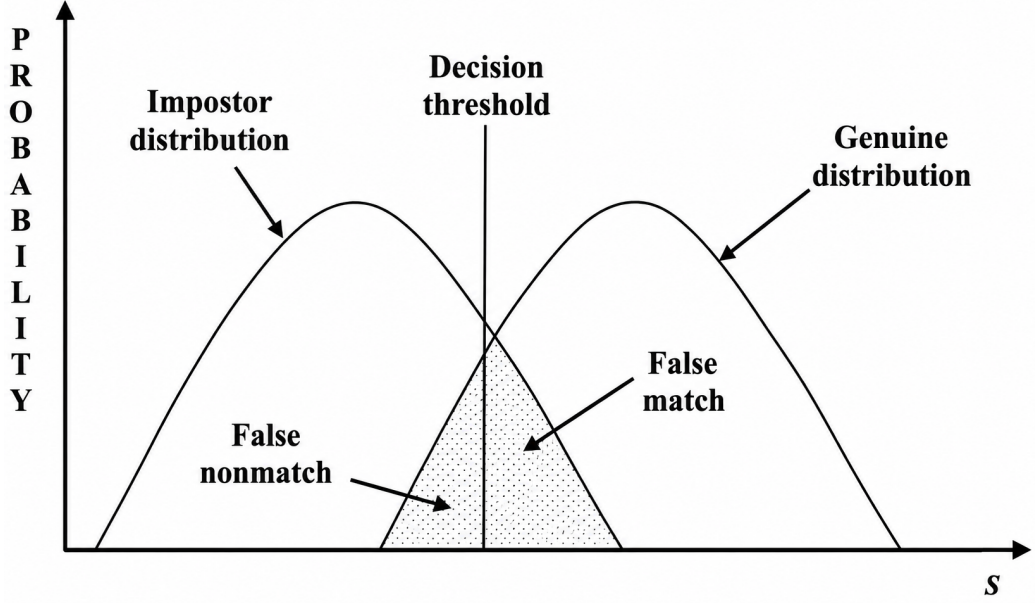


Figure 2.1: Visualisation of Genuine-Impostor distributions. Image taken from [8].

and the False Positive Rate (FPR) [5]:

$$FPR = \frac{FP}{FP + TN}. \quad (2.4)$$

The ROC curve is then constructed by plotting TPR on the y-axis against FPR on the x-axis across all possible thresholds. The Area Under the ROC Curve (AUC) [5] is used to summarise this curve as a single scalar, ranging from 0 (completely wrong classification) to 1 (perfect result).

2.2.3 Matthews Correlation Coefficient (MCC)

The Matthews Correlation Coefficient (MCC) [5] is a metric for evaluating binary classifiers that, unlike ROC AUC, accounts for all four entries of the confusion matrix simultaneously. For a given threshold t , the system produces a binary decision for each pair, which can be mapped to true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The MCC is then computed as follows [5]:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \quad (2.5)$$

The MCC returns a value in the range $[-1, +1]$, where $+1$ corresponds to perfect classification, 0 indicates performance equivalent to random chance, and -1 reflects a completely inverse prediction [5]. The key property of the MCC is that a high score is achieved only when the classifier performs well across all four fundamental rates of the confusion matrix: sensitivity, specificity, precision, and negative predictive value. No other widely used confusion matrix statistic shares this property.

2.3 Stylometry

Stylometry [30, 50] can be defined as a scientific discipline that deals with the analysis and measurement of the style of linguistic expression of an individual or a subset of persons. This recognition relies on the accuracy of various statistical methods, which are based on the knowledge that every author of a text has their own quantifiable way of expression.

Stylometry is part of Behavioural biometry (described in Section 2.1.2). It is often referred to as computational stylistics because it combines traditional linguistic analysis with modern computational techniques to identify unique patterns in the way authors use language. The basic assumption is that authors develop relatively consistent, unconscious writing habits that form their “stylistic fingerprints” [50] – patterns that are difficult to imitate or intentionally change.

2.3.1 History of Stylometry

The historical roots of stylometry date back to the mid-19th century, when the British mathematician and statistician Augustus De Morgan suggested that authors could be identified or recognised using quantifiable features such as word length. A significant impulse came in the 1940s with the work of Udny Yule [30, 41] on the stability of vocabulary patterns regardless of sample size.

A fundamental breakthrough in the field of authorship attribution and authorship verification occurred in 1964, when Frederick Mosteller and David Wallace applied Bayesian statistical methods to resolve long-standing disputes over the authorship of the Federalist Papers [30]. These texts were written during a period of intense debate over the ratification of the United States Constitution (1787–1788), and their authors were three prominent representatives of American political philosophy: Alexander Hamilton, James Madison, and John Jay. However, for more than 150 years, there has been uncertainty about the authorship of some of the essays. With their work, Mosteller and Wallace not only convincingly attributed the disputed texts to James Madison but also laid the foundations for the modern use of statistical and computational methods in quantitative linguistics and authorship attribution.

A significant shift occurred in the second half of the 20th century when stylometric procedures began to be applied in criminalistics. The case of the so-called Unabomber, whose real name was Theodore Kaczynski, is particularly well-known. Between 1978 and 1995, he carried out a series of bomb attacks that claimed three lives and injured more than twenty people. A crucial turning point in the search for the perpetrator was the publication of his manifesto in the newspapers *The Washington Post* and *The New York Times*. The manifesto exhibited distinct linguistic and stylistic features that matched those of Kaczynski’s previous correspondence. His brother, David, contributed significantly to the identification, recognising specific means of expression and formulations. Subsequent linguistic and stylometric analyses confirmed these similarities and enabled a reliable attribution of the text. This case thus became one of the most significant examples of the practical application of stylometry in criminal law and forensic linguistics [41].

2.3.2 Practical Usage of Stylometry

Stylometry is now utilised for various applications across academia, forensics, and digital verification. This analysis is based on linguistic patterns (see Section 2.4.1) to identify and verify authors or detect anomalies.

As a first use case for stylometry, we can consider authorship attribution [50, 13]. This task is particularly important in several fields, including literature and forensics. In literature, stylometry has proven that the novel *A Cuckoo’s Calling*, published by the author *Robert Galbraith*, is, in reality, a pseudonym for the famous writer *J.K. Rowling*. This claim was substantiated through advanced stylometric analysis, comparing this work with works by J.K. Rowling. In a forensic use case, we can mention linking anonymous threads, bomb letters, or cyberstalker messages to suspects. An example might be the *Unabomber* case described in Section 2.3.1.

The next use case for authorship verification we can mention concerns a problem with generative artificial intelligence [29, 2], specifically in the context of large language models (LLMs; see Section 3.1.6). The analysis distinguishes human idiosyncrasies from the uniformity of LLMs. The main attributes that distinguish a person from an LLM can be, for example, *Lexical diversity*, where a human should have a more diverse vocabulary than an AI. Another example might be *Sentence Length*, where the variance in the words used in a sentence is much higher than that of AI. Other attributes that stylometry focuses on might be *Function Words* (like the, of, in, etc.), *syntactic complexity*, or *Semantic Coherence*.

Worth mentioning is the usage of Stylometry in Cybersecurity [2]. The researchers are attempting to profile hackers through their phishing emails or troll posts. With that knowledge, they can track the hackers’ activities online.

2.3.3 Temporal Evolution of Stylometric Characteristics

The assumption that an author’s idiolect remains stable over time is a cornerstone of stylometric analysis [11]. However, empirical research shows that writing style evolves systematically throughout an individual’s lifespan as a result of cognitive development and biological ageing. One of the most consistently observed changes is a gradual decline in lexical richness. This decline is reflected in reduced vocabulary diversity and a lower incidence of hapax legomena², which are often replaced by more frequent use of indefinite nouns and recurring, formulaic expressions.

In contrast, certain aspects of morphological complexity, such as average word length, tend to increase in later works. This pattern suggests a shift toward more formalised or habitual language use rather than a simple reduction in linguistic ability. Importantly, these changes primarily affect content-bearing vocabulary. The use of high-frequency function words (e.g., the, with, and) remains strikingly stable over time, likely because such words are produced largely subconsciously. This stability provides a reliable syntactic baseline that enables accurate authorship attribution across different stages of an author’s life [11].

Moreover, stylistic change over time is sufficiently measurable that pronounced deviations in late-career writing are increasingly used as clinical indicators of neurocognitive decline [11].

2.4 Existing approaches to Stylometry Analysis

The field of stylometry has evolved significantly from subjective literary criticism to a rigorous, data-driven discipline. Contemporary authorship attribution does not rely on a single technique but rather employs a combination of linguistic feature extraction and statistical processing to identify unique writing signatures.

²Hapax legomenon stands for words occurring only once [30].

2.4.1 Stylometric Feature Categories and Analysis Approaches

As mentioned in the Chapter’s introduction, stylometric analysis relies on quantifying a wide range of linguistic features that collectively characterise an author’s writing style. We can divide these individual characteristics into several main parts, each capturing different aspects of linguistic behaviour and the cognitive processes underlying the author’s text production.

Lexical Characteristics

Lexical features [41] represent the foundation of stylometric analysis, as they describe the properties of the vocabulary and the way it is used in the text. Basic metrics, such as the total word count, the number of unique words, and the total character count, provide an initial picture of the text’s scope and complexity. From a psycholinguistic perspective, these indicators reflect the cognitive difficulty of processing and the author’s ability to generate and structure thoughts.

Vocabulary richness can be expressed using various indices. The most commonly used is the Type-Token Ratio (TTR) [39], which is defined as the ratio of unique words to the total number of words. Higher TTR values signal a more diverse vocabulary and a more active use of semantic memory. An alternative approach is Yule’s K [59], published in 1944, which is based on the formula [59] :

$$K = 10^4 \times \frac{\sum i^2 V_i - N}{N^2} \quad (2.6)$$

where V_i denotes the number of words with a frequency of i , and N is the total number of words. Simpson’s Index D, which also quantifies lexical diversity, applies a similar principle.

An important indicator is the Hapax Legomena Rate [39], which is the ratio of words that appear only once in the text. High values of this index indicate a creative, context-specific use of lexis, reflecting the author’s originality and ability to avoid redundancy. This parameter is therefore particularly valuable for assessing vocabulary richness.

Another relevant aspect is the average word length [39, 41], which serves as an indicator of vocabulary sophistication. The use of longer words is generally related to an orientation towards technical or abstract language, while a prevalence of shorter words may signal a tendency towards simplification, for example, under the influence of cognitive stress.

Syntactic Characteristics

While Lexical characteristics represent the foundation of stylometry, syntactic features [39] represent one of the most reliable pillars of stylometry. These features capture the structural organisation of sentences, which should be characteristic of a specific author and often independent of the text topic or genre that the author used. Unlike Lexical features described in subsection 2.4.1, which are easily influenced by thematic content, syntactic features reflect authors unconscious linguistic habits, making them more suitable for authorship attribution.

Some of the measurements include sentence length (the number of words in a sentence), the average number of words per sentence, and part-of-speech (POS) tag distribution³. We call these measurements or features; they focus on *Shallow Syntactic Features* [14]. These metrics are used because they do not require full parsing and can provide useful information

³POS tag distribution describes the percentage of text composed of nouns, verbs, and adjectives.

about the author. Another sign we can collect without full parsing is punctuation frequency. We can collect author-specific use of dashes, semicolons, and exclamation marks based on what we can create as a detectable pattern. In conversations, we can focus on emoticons, which also contribute to creating an author pattern [50, 39].

The most innovative approach in syntactic author attribution focuses on dependency grammar [14]. We can describe it as a formalism representing relations between words, such as a subject depending on a verb. In 2012, Hollingsworth introduced the DepWords format. This format allows encoding each word’s dependency chain from the word to the sentence root as a character string. For example [14]:

- The root verb is #
- The subject that depends on this root verb is #c (c = nsubj relation)
- A noun modified by this subject becomes #ca (a = nn relation)

With this transformation of sentences, we obtain sequences that are analysable like n-grams⁴. Testing this approach with detective novels and 3-grams showed 97.5% accuracy [14].

Semantic Characteristics

While lexical and syntactic features dominate the field, semantic approaches [29] capture the conceptual and thematic dimensions of style. The author’s characteristic choice of concepts evokes the representation of the author’s specific stylistic layer.

In the context of semantic stylometry, word embeddings are often mentioned (embeddings are discussed later in Section 3.1.4). In stylometry, this foundational concept suggests that “a word is characterised by the company it keeps” (Firth 1957), expressing that authors with different semantic preferences or knowledge will use words in systematically different contexts. For stylometry, each text can be represented as an average of its word embeddings, creating a semantic fingerprint [29] that captures not only vocabulary choices but also meaning preferences.

Despite its theoretical promise, semantic stylometry [29] comprises only 5-10% of published stylometry research. Semantic stylometry requires advanced NLP (Natural Language Processing) pipelines, which often come with errors and are often genre or task-dependent.

Comparison of methodological approaches

No single methodology can be considered universally dominant across all of the stylometric tasks [29]. The accuracies reported in this section are summarised in Figure 2.2. Instead, the suitability of a particular approach depends primarily on the specific problem setting, dataset characteristics, and application scenario [29]:

- **Literary texts and long documents** - Lexical methods excel (more than 97% accuracy), but syntactic ones provide better robustness to topic variation and greater interpretability. Semantic methods are often limited unless they are used to analyse the thematic evolution of a given text.

⁴N-grams stand for contiguous sequences of n items from a given sample of text or speech.

- **Short texts** (e.g., emails, conversations, and other brief messages) - For short texts, lexical methods outperform syntactic and semantic methods. A short text doesn't provide sufficient information about the author for syntactic and semantic analysis.
- **Scientific and formal papers** - Syntactic and Lexical analysis perform equally well. While syntactic features provide more accurate results across research topics, lexical analysis fluctuates with subject matter.
- **Cross-domain/cross-genre attribution** - Syntactic features provide superior ($\approx 77\%$ average accuracy) to lexical methods ($\approx 62\%$ average accuracy). This behaviour is based on syntactic methods, representing more habitual author choices, which are less influenced by the topic.
- **AI-generated text detection** - Semantic and syntactic methods perform almost equally ($\approx 86 - 91\%$ average accuracy). In these use cases, semantic methods can detect unnatural coherence.

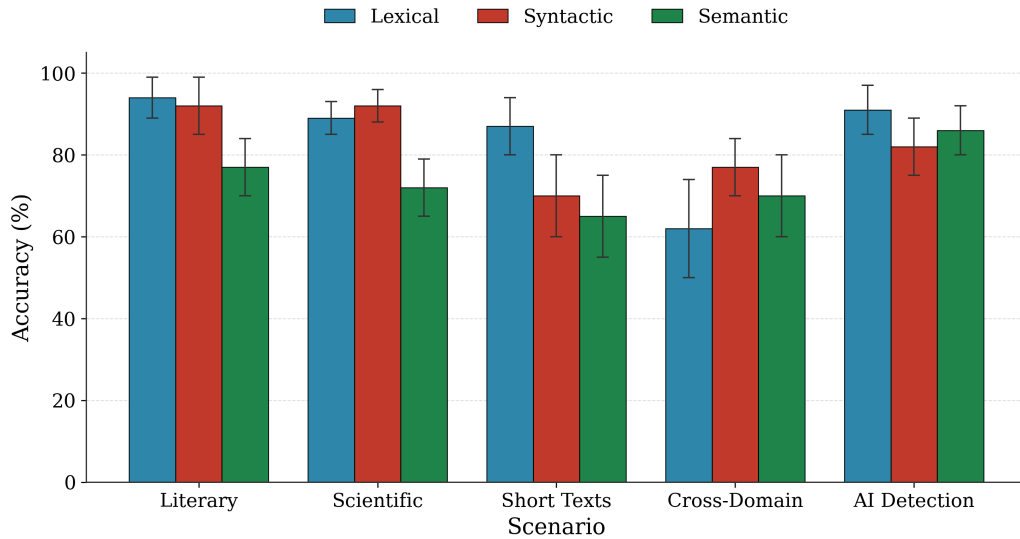


Figure 2.2: Performance of stylometric methods across scenarios. Data taken from [29].

2.4.2 Methodological Frameworks and Algorithms

In the past, people relied, for example, on manual counting of words in a sentence during stylometric analysis and comparing them across multiple texts to identify or verify an author. While transitioning to computational analysis with advanced computational power, multiple algorithms were developed to automate manual work and advance stylometric analysis [50]. Stylometry nowadays is inherently multivariate, analysing hundreds or thousands of dimensions simultaneously. It assumes that while an author might deviate in one dimension (e.g. writing some kind of text in a specific genre), the aggregate of all dimensions (centroid of the author's signature) remains stable.

Burrow’s Delta

Burrow’s Delta (Δ) [22] is a computational metric used for authorship attribution and verification, introduced by John F. Burrows in 2002. The algorithm operates on the concept of nearest neighbours—if we have an anonymous text and a library of texts with known authors, we calculate a numeric *distance* between the anonymous text and all known texts with authors. As a result, we choose the author whose *distance*, or more precisely, the Δ score, is the lowest [29].

The delta score Δ mainly focuses on function words [22] like *the, and, of, to, etc.*. In that case, we can classify the computation of the Δ score as Lexical Stylometry (described in Section 2.4.1). The author chose these words based on the fact that they are highly frequent and “heavily grammaticalized”⁵. Another reason for using these words is that psycholinguistics [22] suggests that they are processed unconsciously in the human brain. In that way, they don’t bring cognitive attention during the writing; it’s more difficult to write, manipulate, or for a forger to imitate successfully.

The process consists of several steps:

- **Vectorisation** - The text is transformed into vectors representing the frequency of each function word.
- **Z-Score Standardisation** - Without proper standardisation, the results would be misleading due to imbalanced variations. To fix this issue, the frequencies in the vectors are converted to z-scores using the Equation:

$$z = \frac{\text{Value} - \text{Mean}}{\text{Standard Deviation}}. \quad (2.7)$$

- **Calculating the delta (Δ)** - When we want to find the Δ distance between document A (anonymous) and document B (known author), Δ score is calculated as the average of the absolute differences between their z-scores :

$$\Delta(\mathbf{A}, \mathbf{B}) = \frac{1}{n} \sum_{i=1}^n |z(\mathbf{A}_i) - z(\mathbf{B}_i)| \quad (2.8)$$

Based on the calculation, we then assign as the final author for the anonymous text as the text whose Δ was the lowest. In 2008, researcher Shlomo Argamon demonstrated that Burrow’s computation of Δ is equivalent to the Manhattan Distance (also known as the L1 norm) applied to z-scaled vectors [22].

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) [1] is a statistical technique used to reduce dimensionality in datasets. It transforms high-dimensional data into a lower-dimensional space while preserving maximum variance. Principal components can be computed using:

$$\mathbf{Z} = \mathbf{X}_c \mathbf{V} \quad (2.9)$$

where $\mathbf{X}_c$ is the centred data matrix and $\mathbf{V}$ contains the sorted eigenvectors. PCA identify the principal components in a data, which will serve as new orthogonal axes derived as linear

⁵Heavily grammaticalized words serve a structural purpose (grammar) rather than a descriptive purpose (meaning).

combinations of original features. While the first computed principal component captures the direction of maximal variance, the second is orthogonal to the first and captures the direction of the second maximal variance.

In the context of Stylometry, PCA is often used for the visualisation of stylometric data. A document-term matrix⁶ may have thousands of dimensions. With PCA, we can transform these correlated variables into a smaller set of uncorrelated variables, or (as mentioned in Section 2.4.2) principal components. By plotting the documents on the first two principal documents, researchers can see clusters of texts from authors. If all cluster from author A are far away of clusters of author B, that provides strong visual evidence of distinct styles. The example can be seen in Figure 2.3, where clusters (*H*, *G*, *B*) are far from each other, which supports the statement that the text was written by three non-identical authors [22, 1].

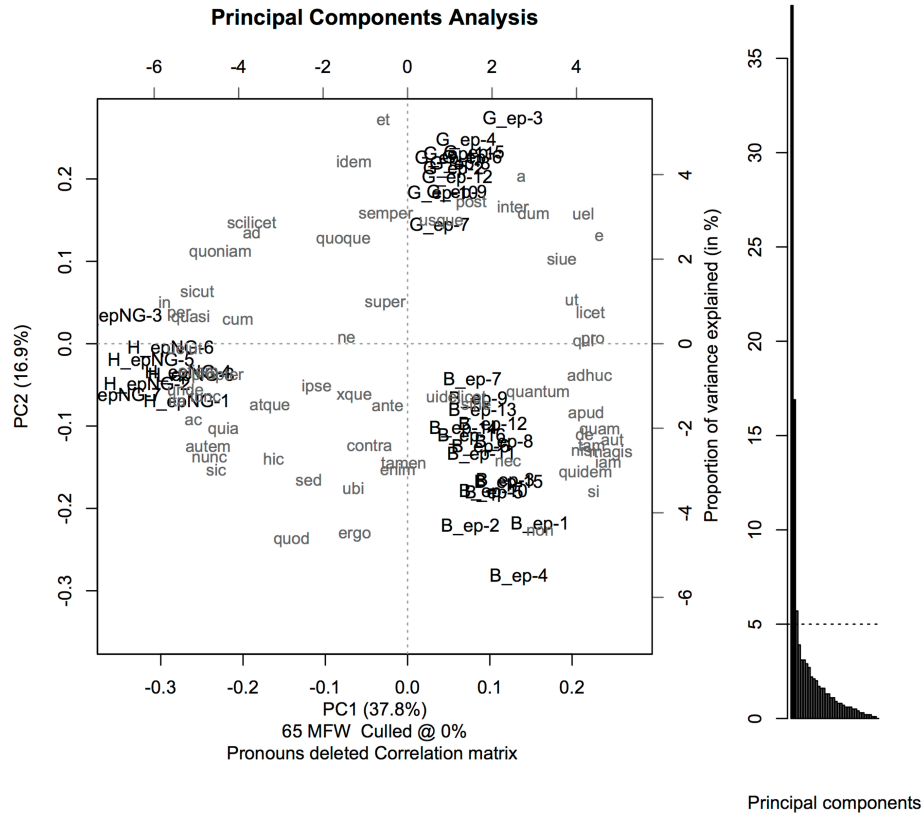


Figure 2.3: PCA plot, first 2 dimensions, contrasting 10,000 lemmatised word samples from three oeuvres. Image taken from [22].

PCA also increases the transparency of the analysis by producing “loadings”. These metrics reveal exactly which words cause the separation between texts. This allows the researchers to explain the results more precisely, such as noting that Author A is distinguished by the use of words like “therefore”, whereas Author B is defined by the use of

⁶Matrix where rows are documents and columns are word frequencies.

“so”. For example, in Figure 2.3, we can see that for author H , the loadings (words) could be *scilicet*, *quoniam*, or *sicut* [22].

Chapter 3

Authorship verification using Stylometry and Machine Learning

In this chapter, we shift our focus to Stylometry in the context of Machine Learning. While Chapter 2 introduced the fundamental concepts and non-machine learning approaches to stylometry analysis—often relying on standard statistical measures or manual feature selection—those methods have inherent limitations in scalability and nuance. By integrating Machine Learning, we can overcome these barriers. This data-driven paradigm allows us to model complex linguistic patterns and dependencies that are often invisible to traditional analysis, effectively pushing the boundary of authorship recognition to a significantly higher level of accuracy and robustness.

In this work, we will focus specifically on *the authorship verification problem* (see Section 2.2). In other words, we will analyse two texts and, at the end of the analysis, decide whether they were written by the same author or by two different authors.

3.1 Machine Learning

Machine learning [21] is a discipline that focuses on two questions: “How can one construct computer systems that automatically improve through experience? And what are the fundamental statistical-computational-information-theoretic laws that govern all learning systems, including computers, humans, and organisations?” [21]. In a broader sense, we aim to enable the computer system to learn automatically from raw data, thereby correctly responding to data that the system hasn’t encountered before [28].

In the context of stylometry, we will use a subset of machine learning called *supervised learning* [28]. For this type of learning, we will use two types of datasets: training and testing. The training dataset will be used to train a model that involves feature extraction and data processing. After training, the trained model and the testing dataset will be used to determine the accuracy and performance of the trained model. For this type of learning, each input in the dataset consists of two parts: the input (used by the model) and the target (the model’s desired output).

For the specific use case of *authorship verification*, the problem can be formulated as learning a decision function that determines whether two texts were written by the same author. Instead of predicting a single author label, the model operates on pairs of texts and learns a mapping that minimises the error in predicting a binary label indicating authorship consistency. Formally, given two input feature vectors $\mathbf{x}_1$ and $\mathbf{x}_2$, the model

learns a function [43]:

$$f : (\mathbf{x}_1, \mathbf{x}_2) \rightarrow Y, \quad (3.1)$$

that outputs a label $Y \in \{0, 1\}$, where $Y = 1$ denotes that both texts originate from the same author, and $Y = 0$ denotes otherwise.

This formulation is particularly suitable for scenarios where the number of potential authors is large or unknown, which is common in forensic applications. Rather than solving a multi-class classification problem, authorship verification [16] reduces the task to a binary decision, effectively addressing the “needle-in-a-haystack” problem. In the modern digital landscape, characterized by the rapid growth of online content such as articles, emails, and blog posts, this approach provides a scalable and practical solution for narrowing down candidate authors.

3.1.1 Support Vector Machine

The Support Vector Machine (SVM) [20] stands as a monumental algorithm in the context of Machine Learning. This approach uses hyperplanes or a set of hyperplanes in high-dimensional space to separate the features or objects across different classes. SVM works by finding a linear classifier that separates data with maximum margin, as can be seen in Figure 3.1. After training, the model is required to satisfy the following constraints. For every training example $(\mathbf{x}_i, y_i)$ with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$ [20]:

$$\mathbf{w}^\top \mathbf{x}_i + b \begin{cases} \geq +1, & \text{if } y_i = +1, \\ \leq -1, & \text{if } y_i = -1. \end{cases} \quad (3.2)$$

These two cases can be combined into the single canonical constraint [20]:

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad \forall i \in \{1, \dots, N\}, \quad (3.3)$$

where $\mathbf{w} \in \mathbb{R}^d$ denotes the weight vector, $b \in \mathbb{R}$ the bias term, and N the number of training samples. Examples for which Equation (3.3) holds with equality are the *support vectors* and lie on the margin boundary.

Unlike traditional neural networks that minimise the error on training data, SVMs use the Structural Risk Minimisation (SRM) [20] principle to minimise the upper bound of expected risk, which improves generalisation.

Soft Margin classifier

In ideal scenarios, data should always be separated by a linear line, making perfect separation impossible. But in the real world, data often contains noise or outliers, which make the perfect separation almost impossible. Additionally, forcing the “hard margin” on noisy data can lead to a decision boundary that is too complex, resulting in overfitting. As the solution for this problem, it’s possible to use *slack variables* [20] which measure the error, allowing specific data points to fall on the wrong side of the margin. In that case, we can edit the original condition in Section 3.1.1 to a simplified constraint [20] :

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad (3.4)$$

where ξ_i denotes a small slack distance, and all other parameters remain identical to those in Equation (3.2).

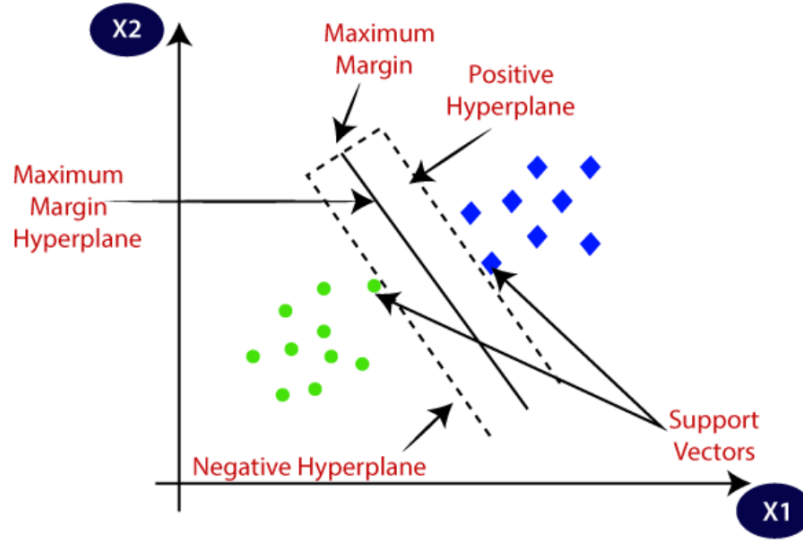


Figure 3.1: Visualisation of data separation using SVM. Image taken from [28].

During training, we aim to minimise the objective functions, which seek to balance two competing goals: maximising the margin and minimising the total slack or error. This is often controlled by the penalty parameter [20] (often denoted as C). A high value for C means errors are more penalised, forcing the model to fit more strictly, while a low value for C allows more slack, resulting in a wider margin.

Kernel functions

Linear classifiers work well for linearly separable data, but many datasets are far from linear and cannot be perfectly separated by a single linear line. In this scenario, it is possible to use *kernels* [28] that will map the original non-linear data into a high-dimensional feature space. Then, we assume that data in higher dimensions will be linearly separable.

Calculating coordinates in high-dimensional space might be very expensive. We might streamline this process by defining *kernel function* [20] $K(\mathbf{x}, \mathbf{x}')$ that computes the dot product of the data in the feature space directly from the input data, without ever needing to perform the explicit mapping. Common kernel functions are [20]:

- Polynomial Kernel:

$$K(\mathbf{x}, \mathbf{x}') = (\langle \mathbf{x}, \mathbf{x}' \rangle + 1)^d \quad (3.5)$$

- Gaussian RBF (Radial Basis Function):

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) \quad (3.6)$$

- Exponential RBF:

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|}{2\sigma^2}\right) \quad (3.7)$$

Choosing the right kernel for data can be a challenging problem that requires multiple runs and testing.

3.1.2 Neural Networks

Artificial Neural Networks (ANNs) [58] are a computing model that abstracts the human brain's neural network from the perspective of information processing. Based on this abstraction, it's possible to create simple models and networks based on different connections between these neurones. The knowledge of ANNs is distributed across the network in the form of *connection weights*. These weights simulate the synaptic connections between the neurones in the brain.

The main element of the ANNs is a neurone, which is composed of these 4 parts [44, 58, 42]:

- **Inputs** - The entrance gate for signals from the beginning of the NN or from previous neurones. In this term, we also include *bias*, a constant that is added to the sum of inputs.
- **Weights** - The strength of the connection between each neurone or input. In this term, we distinguish positive weights as excitatory connections and negative weights as inhibitory connections¹.
- **Activation function** - Decides if the neurone will be activated and the value that will be sent to the following neurones or the output. The activation function is commonly a non-linear function like ReLU, Sigmoid, Softmax².

From a mathematical perspective, if we have inputs for neurone $(x_1, x_2, \dots, x_n)$, weights of neurone $(w_1, w_2, \dots, w_n)$, activation function f , and bias marked as b , we can retrieve the output y of the neurone in Equation [42]:

$$y = f(w_1x_1 + w_2x_2 + \dots + w_nx_n + b) \quad (3.8)$$

3.1.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) [47] are a type of neural network specifically designed to process sequential data. Unlike classical feed-forward neural networks, RNNs handle data where the order of elements matters, allowing them to capture dependencies across time or sequence steps. At each time stamp in the RNN loop, the neural network will take *current input* and *hidden state* from the previous step and produce a new state. These steps can be seen on the left side of Figure 3.2 where x_t represents t input, h represents the hidden state, and z_t represents output.

Long Short-Term Memory

The original RNNs discussed previously struggle to model long sequences due to the vanishing gradient problem. This issue arises during back-propagation through multiple time steps, where gradients progressively shrink and become too small to effectively update the network parameters. As a result, classical RNNs struggle to learn long-term dependencies.

Long Short-Term Memory networks (LSTMs) [12] were introduced to address this limitation. Unlike standard RNNs, LSTMs incorporate an additional memory component

¹Excitatory connections (positive weights) increase the probability of activating the following neurone, while inhibitory connections (negative weights) decrease it.

²The definitions of ReLU, Sigmoid and Softmax functions can be acquired from the following source [44].

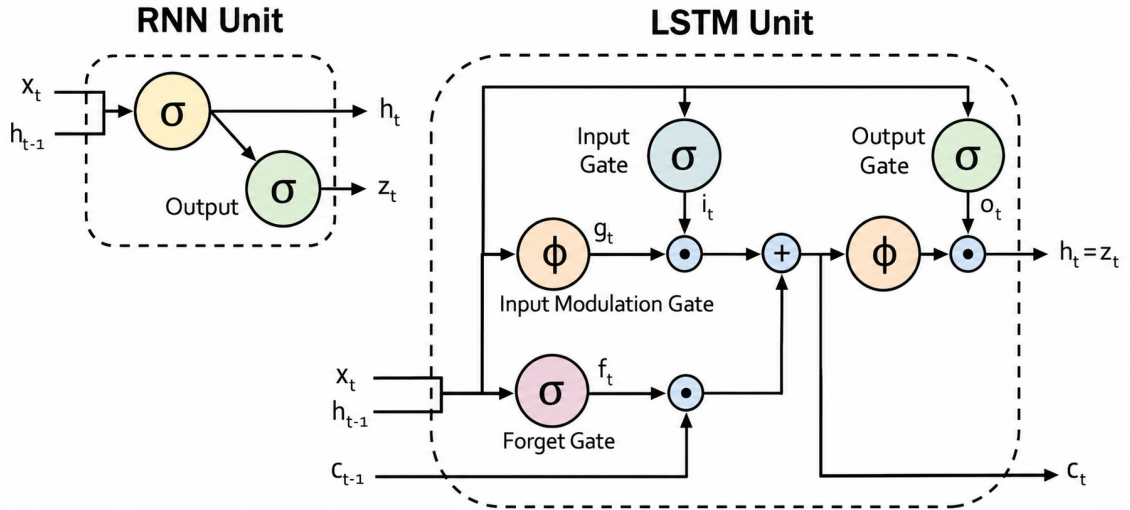


Figure 3.2: Difference between RNN Unit and LSTM unit. Image taken from [54].

known as the *cell state*, which allows information to be preserved over longer periods. Consequently, an LSTM unit has three inputs: *the current input*, *the hidden state*, and *the cell state*, as illustrated in Figure 3.2.

During inference, at each time step, the LSTM unit uses a set of gating mechanisms to regulate the flow of information [12]:

- The **forget gate** determines which information from the previous cell state should be discarded.
- The **input gate** controls which new information is added to the cell state.
- The **cell state** is updated in a way that preserves important long-term information while incorporating relevant new data.
- The **output gate** produces the output of the LSTM unit and updates the hidden state for the next time step.

This gated structure enables LSTMs to maintain stable gradients during training, making them well-suited for tasks that require learning long-term temporal dependencies [12].

3.1.4 Embeddings

Embeddings [4] are a way of representing complex objects, like words, sentences, or documents, in the form of numerical vectors. With this representation, computers and neural networks can work with objects that have meaning rather than with raw information. At the core, embeddings transform human understandable information into a more effective machine form.

The intuition behind embeddings is that similar objects should be closer together in the vector space, while different objects should be further apart. Embeddings reside in a vector space where linear combinations capture analogies, such as *king* − *man* + *woman* ≈ *queen* [4], derived from *word2vec*-style models. This works because similar concepts cluster together, allowing arithmetic to model relational shifts, such as tense or gender.

For searching in the vector space, *cosine similarity* is often used. Cosine similarity measures the angle between two vectors, focusing on directional alignment rather than magnitude differences. The equation for cosine similarity is shown in Equation [4]:

$$\text{similarity}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2} \sqrt{\sum_{i=1}^n b_i^2}} \quad (3.9)$$

where $\mathbf{a}$ and $\mathbf{b}$ are input vectors.

3.1.5 Transformers

Transformers [26] represent a pivotal architecture in deep learning, introduced in the paper “Attention is All You Need” by Vaswani et al. [56], which excels at processing sequential data, such as text, through parallelizable self-attention mechanisms rather than recurrent ones. Transformers are a type of Neural Network that reads a sequence of symbols (similar to RNNs described in Section 3.1.3). The transformer decides which other symbols in the sequence are important by assigning attention weights rather than processing the sequence step by step, as in recurrent models.

Attention mechanism

Paying attention is implemented mathematically with a *self-attention mechanism* [33], which allows every word to look at every other word and assign a relevance score so that the model can understand context and long-range relationships very efficiently. The standard scaled dot-product self-attention mechanism uses the formula [26]:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} \right) \mathbf{V} \quad (3.10)$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are matrices of query vectors, key vectors, and value vectors, respectively, and d_k is the key dimension.

Multi-head Attention

Multi-head Attention (MHA) [26] extends the scaled dot-product attention by running it in parallel across multiple independent “heads”, enabling transformers to capture diverse contextual relationships in sequential data more effectively than single-head attention. Each head can learn to attend to different relationships, while one head might focus on nearby words and another on long-range dependencies [33]. The core computation is based on computing projections for each head_k using the formula [26]:

$$\text{head}_k = \text{Attention}(\mathbf{Q}\mathbf{W}_k^Q, \mathbf{K}\mathbf{W}_k^K, \mathbf{V}\mathbf{W}_k^V) \quad (3.11)$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ have the same definitions as Equation (3.10) and $\mathbf{W}$ are learnt weights. Then, Multi-head attention is calculated by concatenating each head and multiplying by $\mathbf{W}^O$ [33]:

$$\text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) \mathbf{W}^O, \quad (3.12)$$

where $\mathbf{W}^O \in \mathbb{R}^{hd_v \times d_{model}}$.

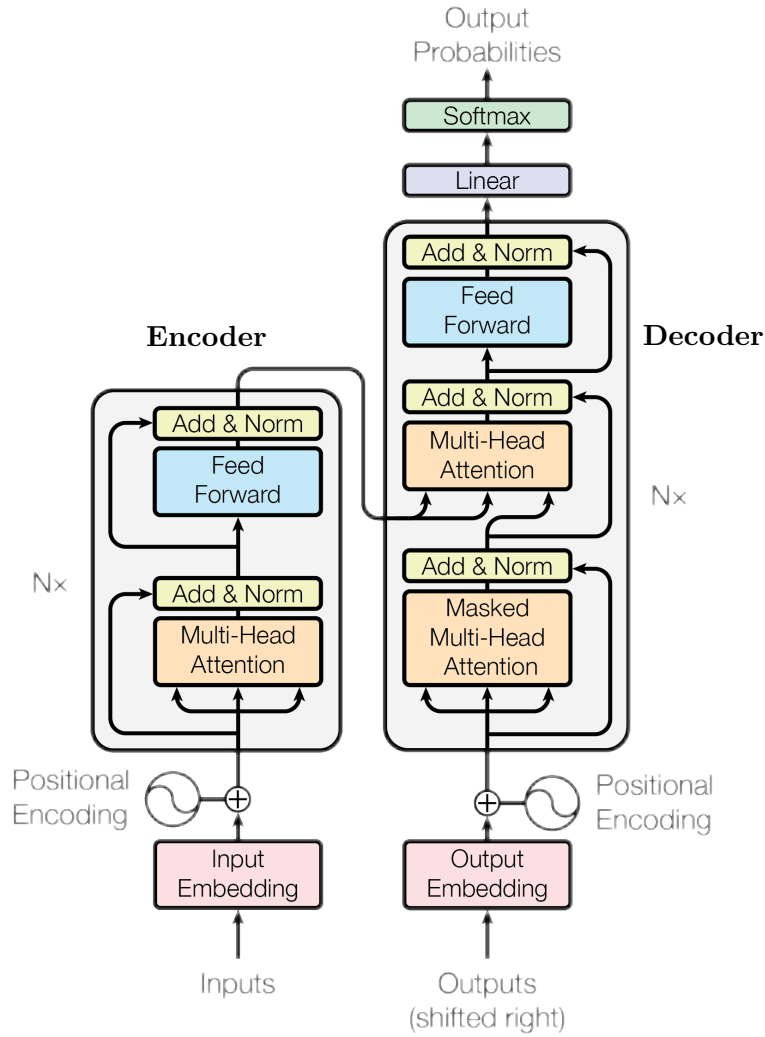


Figure 3.3: Transformer architecture divided into encoder and decoder. Image taken from [33].

Encoder

The encoder [33] is the first part of the Transformer architecture, as shown in Figure 3.3. This part is responsible for converting the input sequence into a set of continuous vector representations. This vector contains semantic information and positional information. The encoder consists of a stack of identical layers, where each layer contains two main sublayers: a Multi-Head Self-Attention layer (described in Section 3.1.5) and a position-wise Feed-Forward layer.

Following the attention sublayer, the output is passed through a Feed-Forward network that is applied independently to each position [33]. This layer introduces non-linearity and helps prevent rank collapse during computation. To ensure stable and efficient training, residual connections are applied around both the attention and feed-forward sublayers. Each residual output is then processed by an Add and Norm layer, which performs layer normalisation. This normalisation reduces inter-layer dependency and mitigates issues such as vanishing gradients.

Decoder

The second part of the classical transformer architecture is the decoder [33]. Structurally, a decoder is similar to an encoder but includes additional mechanisms to support the generation of auto-regressive sequences. The decoder layer applies masked self-attention to previously generated outputs, followed by encoder–decoder attention that allows the Decoder to focus on relevant Encoder representations. The final Decoder outputs are passed through a linear layer and a Softmax function to produce probability distributions over the target vocabulary, and tokens are generated sequentially until the end-of-sequence token is reached.

3.1.6 Large Language Models

Large Language Models (LLMs) [33] are advanced artificial intelligence systems designed to understand, process, and generate human language at scale. These models are trained on massive text corpora and learn complex linguistic patterns, semantic relationships, and syntactic structures. Modern LLMs are predominantly built upon the Transformer architecture (described in Section 3.1.5), which was introduced to overcome the limitations of recurrent and convolutional sequence models, particularly in handling long-range dependencies.

Depending on how the Transformer components are utilised, LLMs can be categorised into several architectural types, each optimised for different language-related tasks [57].

Encoder-only architecture

Encoder-only architectures [57] are primarily focused on language understanding rather than language generation. These models consist of a stack of encoder layers that apply self-attention over the entire input sequence simultaneously. Because the self-attention mechanism is bidirectional, each token can attend to both its left and right context, enabling a deep and holistic understanding of the input text.

This bidirectional contextual modelling makes encoder-only models particularly effective for tasks such as text classification, named entity recognition, sentiment analysis, and question answering [57]. During training, these models are often optimised using objectives such as masked language modelling, where parts of the input are hidden, and the model learns to reconstruct them based on the surrounding context.

Decoder-only architecture

Decoder-only architectures [57] are specifically designed and optimised for text generation. In this configuration, the encoder stack is entirely removed, and the model consists solely of a stack of decoder blocks. These blocks process input tokens in an auto-regressive manner, meaning that each token prediction depends only on previously generated tokens.

Due to their generative nature, decoder-only models are well-suited for tasks such as conversational agents, story generation, code generation, and general-purpose language modelling [57]. They are typically trained using a next token prediction objective, where the model learns to predict the next token in a sequence given all preceding tokens.

This architecture is widely used in contemporary LLMs and scales efficiently to very large parameter counts. Despite lacking an explicit encoder component, decoder-only models are capable of performing a wide range of tasks through prompting and in-context learning.

Encoder-Decoder architecture

The Encoder–Decoder [57] architecture combines the strengths of both previously described approaches. The encoder processes the input sequence to produce a rich contextual representation, while the decoder generates the output sequence by attending both to previously generated tokens and to the encoder’s output.

This architecture is particularly effective for sequence-to-sequence tasks [57] where the input and output differ significantly in structure or length, such as machine translation, abstractive summarisation, and text rewriting. By explicitly separating input understanding and output generation, encoder–decoder models can achieve high performance on complex transformation tasks.

However, this architectural choice comes with increased computational costs. Training and inference are generally more demanding compared to encoder-only or decoder-only models [57], as both components must be executed and coordinated. This makes encoder–decoder models less commonly used in large-scale, general-purpose LLMs, but they are still highly relevant in specialised applications.

BERT

BERT (Bidirectional Encoder Representations from Transformers) [9] is a prominent example of an encoder-only Transformer-based language model [57]. Its key innovation lies in its ability to model bidirectional context by attending to all tokens in the input sequence simultaneously. Unlike earlier language models that processed text in a left-to-right or right-to-left manner, BERT captures contextual information from both directions at every layer. The overview architecture of BERT model can be seen in Figure 3.4.

BERT is pre-trained using two main objectives [9]: masked language modelling and next sentence prediction. These training strategies allow the model to learn deep contextual representations that transfer effectively to a wide range of downstream natural language processing tasks. As a result, BERT and its variants have become foundational models for language understanding applications.

LoRA – Low-Rank Adaptation

LoRA [15] is a machine learning technique used for fine-tuning neural network models (especially transformer models) without updating all parameters during training. The problem with classical training or fine-tuning large models is that this training requires updating billions of parameters, which demands enormous GPU memory and compute; LoRA makes this practical.

For a given weight matrix $\mathbf{W}$ (e.g., in an attention layer), instead of updating $\mathbf{W}$ directly, LoRA freezes $\mathbf{W}$ and learns a low-rank decomposition of the update [15]:

$$\mathbf{W}' = \mathbf{W} + \Delta\mathbf{W} = \mathbf{W} + \frac{\alpha}{r}\mathbf{B}\mathbf{A}, \quad (3.13)$$

where $\mathbf{B} \in \mathbb{R}^{d \times r}$ and $\mathbf{A} \in \mathbb{R}^{r \times k}$ for a given rank $r \ll \min(d, k)$, and α is a scaling hyperparameter that controls the magnitude of the adaptation. Matrix $\mathbf{A}$ is initialised randomly, and $\mathbf{B}$ is initialised to zero, ensuring that $\Delta\mathbf{W} = 0$ at the start of training so the adapted model is initially identical to the pre-trained model. During training, only $\mathbf{A}$ and $\mathbf{B}$ are updated while $\mathbf{W}$ stays frozen.

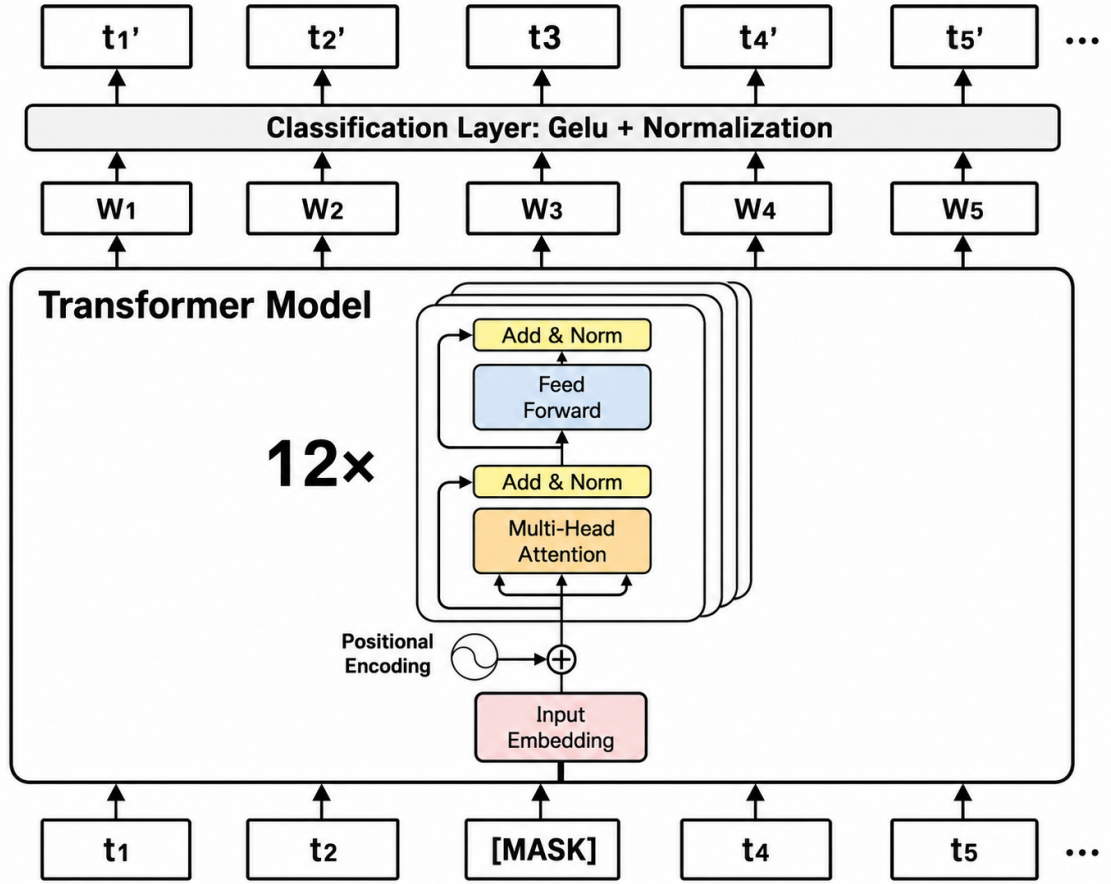


Figure 3.4: BERT masked-language model with token and positional embeddings, stacked transformer encoder layers, output representations, and a GELU-normalised classification head for predicting masked tokens. Image taken from [25].

The hypothesis is that weight updates during fine-tuning have a low intrinsic rank – the meaningful change lives in a much lower-dimensional subspace than the full matrix. LoRA exploits this by only learning that subspace [15].

3.1.7 Loss functions

For our work, we employed several loss functions chosen to match both the model architecture and the desired network output. Specifically, we rely on three loss formulations: Cross-Entropy Loss, Focal Loss, and Supervised Contrastive Loss.

Cross-Entropy

Cross-entropy [27] is one of the most widely used loss functions for classification tasks in machine learning. It measures the discrepancy between the model’s predicted probability distribution and the true distribution of the target labels. The objective of training is to minimise this discrepancy, thereby encouraging the model to assign high probability to the correct class. Formally, for a classification problem with C classes, the cross-entropy loss is defined:

$$\mathcal{L}_{\text{CE}} = - \sum_{i=1}^C y_i \log(p_i), \quad (3.14)$$

where y_i represents the ground truth distribution, typically encoded as a one-hot vector, and p_i denotes the predicted probability of class i produced by the model. In this setting, the loss penalises confident but incorrect predictions more heavily than less certain ones. In the common case of one-hot-encoded labels, the loss simplifies to the negative log of the predicted probability for the correct class. This formulation ensures that the model is encouraged not only to predict the correct class, but also to do so with high confidence.

Focal Loss

Focal loss [32] is an extension of the standard Cross-Entropy loss that introduces a mechanism for emphasising hard-to-classify samples during training (for our use case, semantically distant texts from the same author). While it was originally proposed to address class imbalance in dense object detection tasks, its formulation is general and can also be beneficial in other classification problems where samples exhibit varying levels of difficulty.

In the context of authorship verification, even when the dataset is balanced, the task inherently contains both easy and hard examples. Standard cross-entropy treats all samples equally, potentially leading to insufficient focus on challenging cases. Focal Loss addresses this limitation by reducing the contribution of well-classified examples and emphasising those that are more difficult to classify.

Formally, Focal Loss modifies the cross-entropy loss by introducing a modulating factor, as defined:

$$\mathcal{L}_{\text{FL}}(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t), \quad (3.15)$$

where p_t denotes the predicted probability of the true class, $\gamma \geq 0$ is the focusing parameter, and $\alpha_t \in [0, 1]$ is a weighting factor.

Supervised Contrastive Loss

Supervised Contrastive Loss is a metric learning objective that extends contrastive learning by leveraging label information to explicitly structure the embedding space. Unlike standard classification loss functions such as Cross-Entropy, which learn decision boundaries in the output space, Supervised Contrastive Loss directly optimizes the representation space by pulling samples of the same class closer together while pushing samples from different classes apart [27].

Given a batch of N samples, each sample i is associated with an embedding vector $\mathbf{z}_i \in \mathbb{R}^d$, typically normalized such that $\|\mathbf{z}_i\|_2 = 1$. For each anchor sample i , let $P(i)$ denote the set of indices of all other samples in the batch that share the same class label as i , excluding i itself. The loss is then defined [27]:

$$\mathcal{L}_{\text{SupCon}} = \sum_{i=1}^N \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_p / \tau)}{\sum_{a \neq i} \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)}, \quad (3.16)$$

where $\tau > 0$ is a temperature parameter that controls the concentration level of the distribution. The dot product $\mathbf{z}_i \cdot \mathbf{z}_j$ corresponds to cosine similarity (see Equation (3.9)) due to the normalisation of embeddings [27].

This formulation encourages the model to maximise similarity between an anchor and all positive samples (same class) while minimising similarity to all other samples (negatives) within the batch. Unlike Triplet Loss, which considers only a single positive-negative pair at a time, Supervised Contrastive Loss utilises multiple positives and negatives simultaneously, leading to more stable and efficient training [27].

3.2 Recognition based on SVM

Recent research from 2023 onward has incorporated SVM (described in Section 3.1.1) in authorship verification, often as a baseline or in hybrid models with dimensionality reduction and stylometric features. These newer solutions address challenges such as high-dimensional data and short texts, demonstrating SVM’s continued relevance despite advances in deep learning.

The study from Solórzano et al.(2015) [49] uses multiple one-class SVMs. Each one SVM is trained on a different feature category (e.g. lexical, POS, surface) with a kernel tuned to that type. The outputs from these SVMs are then fed into a logistic regression meta-classifier that automatically weights feature categories. This approach was evaluated on PAN-2015 dataset, with a good results but it doesn’t reach the top-ranking results [49].

The work by Corbara and Moreo (2024) [6] investigates improving authorship verification robustness by training classifiers on synthetic adversarial documents designed to imitate a target author’s style. The authors evaluate SVM as one of the principal classical classification methods alongside neural approaches such as convolutional neural networks. In this framework, SVM serves as a strong traditional baseline for comparison with modern deep learning models. To improve robustness, the study introduces data augmentation techniques that generate more challenging negative verification pairs during training. Their experiments show that SVM remains competitive in stylometric authorship verification, particularly when combined with carefully engineered textual features. Depending on the dataset and adversarial attack scenario, the reported SVM accuracy ranges approximately from 0.75 to 0.88 [6].

Another relevant work is Huang et al. (2024) [17], which addresses AI authorship verification as a binary classification problem aimed at distinguishing human-written texts from AI-generated ones. The proposed method combines fine-tuned authorial language models with a Support Vector Machine classifier that performs the final verification decision using perplexity-derived feature vectors. Unlike traditional stylometric approaches based solely on handcrafted features, this framework integrates language-model-based semantic representations with classical SVM classification. The authors report that their SVM-based verification stage achieves strong performance, reaching a ROC-AUC of 0.979 and outperforming all baseline systems evaluated in the study [17].

3.3 Recognition based on LSTM

LSTM-based solutions for authorship verification effectively model sequential dependencies in stylometric features, surpassing static classifiers like SVM on tasks with variable-length inputs. These models utilise LSTM’s gating mechanisms to retain long-range patterns in authorial style, as validated in benchmarks such as the PAN corpora. Recent syntactic RNN variants further highlight this superiority.

Douglas Bagnall’s (2015) [3] work, *Author Identification using Multi-Headed Recurrent Neural Networks*, introduced a neural approach to authorship verification in the PAN 2015 competition. The proposed method employs a multi-headed recurrent neural network with a shared recurrent hidden layer for learning general character-level language patterns and separate output heads modelling individual authors’ writing styles. Operating at the character level, the system avoids reliance on handcrafted stylometric features and instead learns stylistic representations directly from raw text. For verification, the model evaluates how well an unknown text matches the learnt character distribution of a candidate author compared to alternative author models. This approach captures subtle sequential stylistic traits, such as punctuation and spelling patterns, and achieved the best overall performance in the PAN 2015 shared task [3].

3.4 Recognition based on Transformers

Transformers excel in authorship attribution by leveraging contextual embeddings to capture stylistic signals, such as vocabulary patterns, punctuation, and syntax. Four key approaches illustrate this progression from simple fine-tuning to contrastive and generative methods.

The work by Manolache et al. (2021) [35] describes how pre-trained encoders such as *BERT* and *RoBERTa* can be adapted to the authorship verification task via transfer learning. The problem is formulated as a binary classification task, where the goal is to determine whether two texts were written by the same author. The approach employs a cross-encoder architecture, in which both input texts are processed jointly. Experimental results demonstrate competitive performance on PAN benchmarks. Specifically, the authors report performance of approximately 90–92% in same-topic settings, while performance drops to approximately 70–80% in cross-topic scenarios [35].

Another work that employs a transformer-based approach is the study by Tyo et al. (2021) [55]. Instead of jointly encoding both texts, this approach learns a shared embedding space in which embeddings of texts written by the same author are close, while embeddings of texts from different authors are far apart. The architecture is based on a Siamese network consisting of two identical BERT encoders with shared weights. Cosine similarity is used to compare the resulting embeddings, and the model is trained using contrastive loss. The authors also employ hard negative mining, where texts are embedded and the most similar samples from different authors are selected as negative pairs. This encourages the model to learn discriminative stylistic features rather than relying on superficial semantic similarity. The reported results indicate an accuracy of approximately 80–88% on standard benchmarks. In more challenging cross-topic scenarios, the accuracy decreases to approximately 75–85%, with a smaller performance drop compared to cross-encoder BERT models, indicating better generalisation [55].

The work by Li et al. (2025) [31] investigates how pre-trained transformer encoders, such as *BERT*, can be effectively adapted to the authorship verification task. The authors focus on simple yet robust strategies rather than complex architectures. The task is formulated as a pairwise similarity problem, emphasizing the use of embeddings and similarity measures. Each text is encoded independently, and cosine similarity is used to determine whether two texts were written by the same author. Instead of heavy training, the authors employ only light fine-tuning and emphasize the importance of selecting an optimal decision threshold, which is tuned on a validation set. Experimental results show that this

simple approach achieves competitive performance compared to state-of-the-art methods, with accuracy reaching approximately 80–90% [31].

As mentioned in Section 3.2, the work by Huang et al. (2024) [17] employs, in addition to an SVM classifier, GPT-style causal language models. In this approach, each language model is fine-tuned on texts from a specific author. The system computes perplexity scores of a given text under multiple author-specific models, and these scores are subsequently used as input features for the SVM classifier. The authors show that combining multiple author-specific language models with perplexity-based features significantly improves robustness and overall performance.

Chapter 4

Data Preparation and Establishing a Preliminary Experiment

The current research in Author Verification based on stylometry is quite advanced, and the results from those solutions are very good. The problem this issue faces is semantic. According to the solutions we described previously in Chapter 3, the authors are strictly focusing on architecture design and feature extraction using Deep Neural Networks (DNNs), which is a standard procedure in the machine learning process. On the other hand, the problem arises when the Neural Network primarily learns the semantics of the input texts.

4.1 Data collection

For reliable authorship verification, it is essential to employ datasets that cover a broad spectrum of textual genres. Such datasets should include a large number of distinct authors, with each author represented by as many samples as possible. This diversity ensures robust model training and improves the generalisation capability of the resulting system.

Based on these requirements, six publicly available datasets were selected for the authorship verification task. These datasets encompass texts from multiple domains, including blogs, literary works of various genres, and short-form messages. Each text sample is accompanied by annotations specifying its corresponding author, which enables supervised learning and evaluation.

4.1.1 Blogtext dataset

The Blog Texts dataset consists of a collection of blog posts from 2004, authored by a large number of individuals writing on a wide range of topics [46]. This diversity makes it a valuable resource for authorship verification tasks.

The dataset contains 681,284 texts from 19,320 unique authors, making it the largest dataset used in this work. The average text length is approximately 200 words, providing relatively concise samples compared to other datasets considered.

4.1.2 Reuters C50 Dataset

The Reuters C50 dataset is a widely used benchmark for authorship attribution and authorship verification tasks. It consists of English news articles collected from the Canadian news agency Reuters [34].

The dataset is relatively small in scale, comprising 50 authors designated for training and testing. Each author is represented by 50 text samples for training and 50 text samples for testing, resulting in a total of 5,000 documents.

4.1.3 IMDB62 dataset

Compared to the C50 dataset, this dataset reflects more realistic scenarios, as it consists of movie reviews collected from the IMDB62 dataset. The texts are primarily written in English and exhibit greater variability in style and content [48].

The dataset comprises 61,987 texts from 62 distinct authors. While this relatively large number of samples per author facilitates the construction of positive pairs, it also increases the complexity of negative pair mining, as distinguishing between authors becomes more challenging in such a dense representation space.

4.1.4 News Category dataset

The News Category dataset is primarily designed for news topic classification tasks. However, it also contains metadata identifying the author of each article, which enables its use for authorship verification. The original dataset provides only article links. Therefore, a web scraping process was required to retrieve the full textual content [37].

After scraping, the dataset comprised 83,455 documents authored by 1,004 distinct authors. A notable characteristic of this dataset is the relatively large average document length, approximately 3,200 words per article. One possible approach to address this would be to split long documents into smaller segments. However, given the average of 83 documents per author, such a strategy would not significantly increase the diversity of writing styles within the dataset and was therefore not adopted.

However, news articles may contain additional noise for stylometric analysis, as their final form can be affected by editorial corrections and publication guidelines. Consequently, the extracted stylistic patterns may not always reflect only the individual author’s writing style.

4.1.5 PAN20 dataset

The PAN20 dataset provides a large collection of text pairs specifically designed for authorship verification tasks. It consists of cross-topic and cross-domain texts, primarily in English, making it suitable for training the model and evaluating its robustness under varying semantic conditions [23].

The dataset contains more than 275,000 pairs, with relatively long documents averaging approximately 21,000 words. As in previous cases, document splitting was not applied for similar reasons related to dataset diversity and structure.

However, since the dataset is already pre-constructed as author-labelled pairs, it does not align with the proposed dataset creation pipeline, which operates on individual texts and dynamically constructs pairs. Therefore, this dataset was excluded from the final dataset generation process.

4.1.6 PAN21 Dataset

The PAN21 dataset is a successor to the PAN20 dataset and represents a challenging and robust benchmark for authorship verification tasks. It consists of multilingual and cross-

domain texts, designed to evaluate a model’s ability to capture author-specific writing style independently of language [24].

The dataset contains approximately 20,000 text pairs, with document lengths comparable to those in the PAN20 dataset. In this work, the PAN21 dataset is used exclusively for evaluation as a secondary test set for comparing with different approaches. Consequently, it is not included in any stage of the dataset construction or splitting process described in Section 4.2.

4.2 Data preparation

As a result of the data collection process described in Section 4.1, the obtained datasets were available in multiple formats and structural variations. Therefore, a preprocessing step was required to ensure consistency across all sources. For each dataset, a dedicated data connector was implemented to transform the data into a unified representation using the Pandas DataFrame format, which provides efficient data manipulation and processing capabilities.

From each dataset, two key attributes were extracted: the textual content and the corresponding author identifier. These data are subsequently preprocessed into pairs for the authorship verification task.

4.2.1 Pairs mining

An important improvement introduced in this work is the incorporation of advanced pair mining strategies. In many existing approaches (as described in Chapter 3), training pairs are typically generated through random sampling. While this method preserves the overall distribution of matching and non-matching author pairs, it does not explicitly control for the semantic relationship between texts.

As a consequence, the model may implicitly rely on semantic similarity as a primary signal for authorship verification. In practice, this means that texts with similar topics or vocabulary are more likely to be classified as originating from the same author. Although this behaviour can lead to correct predictions in cases where an author consistently writes about similar subjects, it introduces a significant limitation in more realistic scenarios.

Authors often produce texts across a wide range of topics and styles. When semantically dissimilar texts written by the same author are presented, a model trained on randomly sampled pairs may struggle to correctly associate them. This can result in an increased number of false negative predictions, where texts from the same author are incorrectly classified as belonging to different authors.

To address this limitation, the proposed approach focuses on more informed pair selection, aiming to reduce the model’s reliance on semantic similarity and instead encourage the learning of deeper stylistic and author-specific features. This leads to improved robustness, particularly in cases where semantic content varies significantly across texts written by the same author.

The full illustrated pipeline used for pairs mining can be seen in Figure 4.1.

Embedding of texts

To achieve this, a lightweight Transformer-based encoder model, *all-MiniLM-L6-v2*, from the Sentence Transformers library was utilised. This model maps sentences or paragraphs

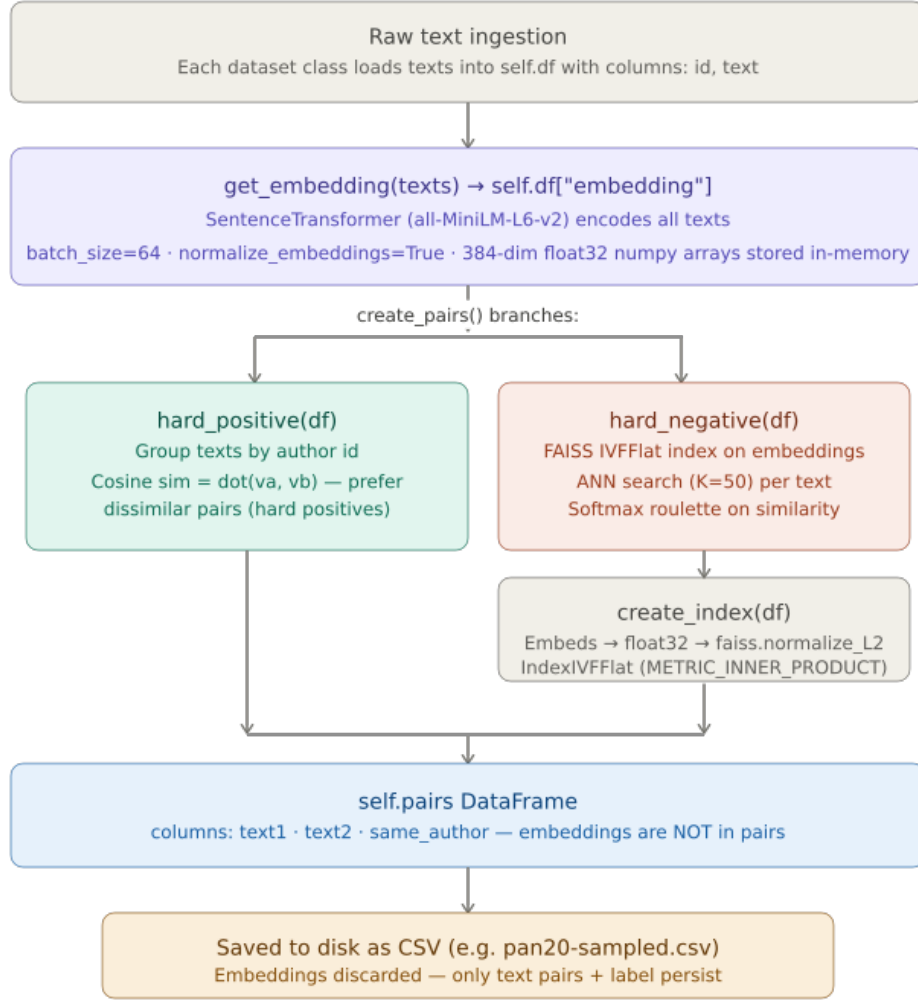


Figure 4.1: Data processing pipeline describing the pair-mining procedure.

into a 384-dimensional dense vector space, enabling efficient comparison of textual semantics. It has been trained and fine-tuned on a large-scale dataset of approximately one billion sentence pairs using contrastive learning objectives [45].

The model was selected primarily due to its strong balance between embedding quality and computational efficiency. In particular, its low inference latency makes it well-suited for processing large-scale datasets, where both speed and representational performance are critical.

The unified DataFrame was subsequently processed by the embedding model using a batch size of 64. The resulting embeddings were normalised using L2 normalisation defined as:

$$\hat{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2} = \frac{\mathbf{x}}{\sqrt{\sum_{i=1}^n x_i^2}} \quad (4.1)$$

where $\mathbf{x}$ denotes the input vector. The computed embeddings were then temporarily stored within the DataFrame for further processing. However, they were not serialised and existed only in memory during execution.

Hard Positive Pair Mining

To obtain hard positive pairs, the dataset is processed at the level of individual sub-datasets and authors. For each author with at least two available texts, embeddings are grouped according to the author identifier within the DataFrame.

From these groups, candidate pairs are sampled, and their similarity is computed using cosine similarity defined in Equation (3.9). Given that all embeddings are L2-normalised, the cosine similarity between two vectors $\mathbf{v}_a$ and $\mathbf{v}_b$, where θ denotes the angle between them, reduces to their dot product (the same idea as in Supervised Contrastive Loss, see Section 3.1.7):

$$\mathbf{v}_a \cdot \mathbf{v}_b = \sum_{i=1}^n (v_a)_i (v_b)_i = \|\mathbf{v}_a\|_2 \|\mathbf{v}_b\|_2 \cos \theta \quad (4.2)$$

Subsequently, pairs are selected from the candidate set with probability proportional to their distance, defined as one minus the cosine similarity defined in Equation (3.9). This sampling strategy increases the likelihood of selecting pairs (pairs of embeddings $\mathbf{v}_a, \mathbf{v}_b$) that are more dissimilar in terms of their semantic representation:

$$d(\mathbf{v}_a, \mathbf{v}_b) = 1 - \cos(\mathbf{v}_a, \mathbf{v}_b) = 1 - \frac{\mathbf{v}_a \cdot \mathbf{v}_b}{\|\mathbf{v}_a\|_2 \|\mathbf{v}_b\|_2} \quad (4.3)$$

As a result, texts written by the same author but exhibiting greater stylistic or topical variation are more likely to be selected. These pairs are referred to as *hard positives*, as they present a more challenging learning signal and encourage the model to focus on deeper author-specific characteristics rather than surface-level semantic similarity.

FAISS Index Construction

Facebook AI Similarity Search (FAISS) is a library for efficiently storing and retrieving high-dimensional embeddings using nearest-neighbour search. Instead of performing a brute-force comparison of a query vector against all vectors in the database, which has a time complexity of $O(N)$, FAISS constructs specialised index structures that significantly accelerate the search process.

Given the scale and dimensionality of the embeddings used in this work, employing a standard `IndexFlatIP` index on a GPU was not optimal. Although it benefits from parallelisation, it still relies on a brute-force search strategy and incurs additional overhead from data transfers between host and device memory.

For this reason, the `IndexIVFFlat` structure was selected as a more suitable alternative. Despite operating on the CPU, it enables a more efficient search through approximate nearest neighbour techniques. The Inverted File (IVF) index partitions the embedding space into n_{list} Voronoi cells using k-means clustering. Each cell is represented by a centroid, and individual embeddings are assigned to their nearest centroid.

During query time, instead of scanning the entire dataset, the search is restricted to a subset of cells whose centroids are closest to the query vector. This significantly reduces computational costs at the expense of a controlled approximation error, representing a trade-off between accuracy and speed.

The number of Voronoi cells, n_{list} , was set to $\sqrt{N}$, where N denotes the total number of embeddings. This heuristic provides a practical balance between search efficiency and the stability of the k-means clustering process.

Hard Negative Pair Mining

Hard negative pairs are constructed using the `IndexIVFFlat` FAISS index described in Section 4.2.1, which is built separately for each dataset. For each sample, a text instance is randomly selected, and its nearest neighbours are retrieved from the index.

Specifically, the top $K = 50$ nearest neighbours are obtained based on cosine similarity in the embedding space. The value of K was chosen as a trade-off between computational efficiency and sample diversity. This neighbourhood size provides sufficiently diverse candidates for informative sampling, while avoiding overly distant samples that would represent trivial (easy) negatives. The process of pair mining and choosing the pairs is visualised in Figure 4.2.

After retrieving the nearest neighbours, all samples originating from the same author as the query text are excluded. This masking step is essential to prevent incorrectly labelling same-author pairs as negatives, which would otherwise introduce noise and degrade the training process.

Following this filtering, the remaining candidates are processed using the core sampling mechanism referred to as *softmax roulette*. Each candidate i is assigned a probability p_i based on its similarity score s_i , computed using a softmax function with temperature $\tau = 0.2$, as defined below:

$$p_i = \frac{\exp(s_i/\tau)}{\sum_j \exp(s_j/\tau)} \quad (4.4)$$

This strategy increases the likelihood of selecting semantically similar and therefore more challenging negative samples, while still maintaining stochasticity in the selection process.

For numerical stability, a standard max-subtraction technique is applied prior to the softmax computation. Specifically, the maximum similarity score is subtracted from each s_i , by following Equation:

$$s_i = s_i - \max_j s_j \quad (4.5)$$

Finally, the dataset was divided into three subsets: training, validation, and testing in a ratio of 8:1:1. To ensure a proper evaluation protocol, the dataset construction process enforces an author-disjoint split, meaning that authors present in the training set do not appear in either the validation or testing sets. This prevents information leakage and ensures that the model is evaluated on previously unseen authors.

Additionally, any class imbalance introduced after the splitting process is addressed by balancing the number of positive and negative samples within each subset. This step ensures stable training dynamics and fair evaluation across all partitions.

4.2.2 Dataset Overview

From the pair mining process, a dataset was constructed and subsequently divided into three subsets. In total, 679,354 pairs were generated, comprising 643,226 training pairs, 19,066 validation pairs, and 17,062 test pairs.

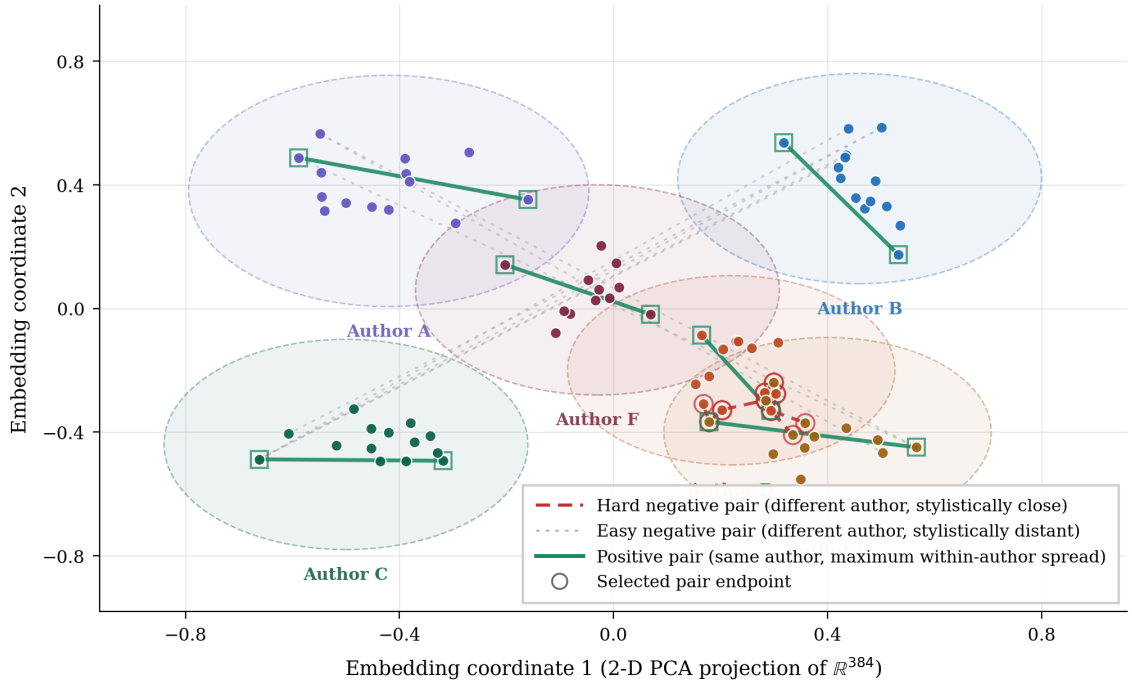


Figure 4.2: Hard negative mining in the SentenceTransformer embedding space (all-MiniLM-L6-v2, $\mathbb{R}^{384}$, shown as 2-D PCA projection).

The overall number of pairs is considered sufficient for the purposes of this study. Since the evaluated models are pre-trained and already possess a strong understanding of natural language, the objective is not to learn linguistic structures from scratch, but rather to distinguish between authors and identify stylistic patterns within the data.

An important characteristic of the constructed dataset is the intentional inclusion of semantically challenging cases. Specifically, the dataset contains (i) positive pairs that are semantically distant and (ii) negative pairs that are semantically similar. This design reduces the model’s reliance on semantic similarity and encourages it to focus on deeper stylistic features.

An example of a **positive pair** (same author, semantically distant) is shown below (full version of the text can be seen in Appendix B.1):

Text 1: Anglo-Dutch food-to-detergent group Unilever Plc, reporting third quarter results next Friday, is likely to have suffered from restructuring costs for its North American industrial cleaning business Diversey and from weak ice cream sales in a drab European summer, analysts said. ...

Text 2: British cider group Matthew Clark said on Tuesday fashionable "alcopop" drinks had helped to wipe a third off the sales of its premium brands -- Diamond White, K and Babycham -- in the first five months of the financial year. ...

Although both texts are written by the same author, they differ in topic and vocabulary, making them semantically distant.

Conversely, an example of a **negative pair** (different authors, semantically similar) is presented below (full version of the text can be seen in Appendix B.2):

Text 1: A festive calm settled over Hong Kong on Thursday as people prepared to welcome the Year of the Ox, the territory’s last Lunar New Year celebration under British rule. ...

Text 2: China has revived a call by Communist Party chief Jiang Zemin to wipe out poverty before the end of the century in a bid to placate its backward hinterland, analysts said. ...

In this case, both texts are semantically related, as they discuss similar geopolitical contexts, yet they originate from different authors. Such examples illustrate the limitations of relying solely on semantic similarity and highlight the importance of modelling author-specific stylistic features.

4.3 Preliminary Experiment Setup

Many existing approaches leverage generally available LLMs for authorship verification tasks. These models are pre-trained on large-scale corpora and are capable of capturing complex linguistic patterns, making them well-suited for distinguishing stylistic characteristics of text.

In this section, several LLMs of varying architectures, sizes, and origins are evaluated. Their performance is assessed through inference on the two test subsets constructed in this work, allowing for a comparative analysis of their effectiveness in the authorship verification task.

The motivation behind this approach is to highlight the role of semantic similarity in authorship verification. Specifically, it aims to demonstrate that models tend to associate texts that are semantically similar more easily, while encountering greater difficulty when comparing texts that are semantically distant, even if they originate from the same author.

4.3.1 Experiments models

For this experiment, the models were selected based on their availability, size, and reported performance on general benchmark tasks. The selection was designed to include both smaller and larger models in order to provide a balanced comparison and to analyse how model scale influences performance in the authorship verification task.

Qwen3

Alibaba Cloud’s Qwen3 [52] is a family of open-source large language models based on the decoder-only Transformer architecture (described in Section 3.1.6). The Qwen3 family includes both dense and Mixture-of-Experts (MoE) variants, with model sizes ranging from 0.6B to 235B parameters, enabling flexible trade-offs between computational efficiency and performance.

A key innovation introduced in Qwen3 is the integration of two operational modes within a single model: a *thinking mode*, designed for complex multi-step reasoning, and a *non-thinking mode*, optimized for fast and efficient inference [52]. This unified framework allows the model to dynamically adapt its computational behavior depending on the task, effectively balancing latency and reasoning capability. Additionally, the concept of a *thinking budget* enables explicit control over the computational resources allocated during inference.

Another important characteristic of Qwen3 is its strong multilingual capability, expanding support from 29 to 119 languages and dialects. This significantly improves cross-lingual understanding and makes the model applicable to diverse textual domains.

For the purposes of this work, the smaller dense variants from the Qwen3 family, namely *Qwen3:1.7B* and *Qwen3:4B*, are utilised. These models are designed to inherit knowledge from larger-scale variants while maintaining significantly lower computational requirements. As a result, they provide a favourable trade-off between performance and efficiency, making them particularly suitable for deployment in resource-constrained environments and for fast inference scenarios.

Table 4.1: Comparison of Qwen3 small-scale models (1.7B and 4B) with open-source baselines (adapted from [52]).

Benchmark	Gemma-3-1B	Qwen2.5-1.5B	Qwen3-1.7B	Qwen2.5-7B	Qwen3-4B
MMLU	26.26	60.90	62.63	74.16	72.99
MMLU-Pro	9.72	28.53	36.76	45.00	50.58
BBH	28.13	45.10	54.47	70.40	72.59
GSM8K	2.20	68.54	75.44	85.36	87.79
MATH	3.66	35.00	43.50	49.80	54.10
EvalPlus	8.98	44.80	52.70	62.18	63.53
MBPP	9.20	43.60	55.40	63.40	67.00
MGSM	1.74	32.82	50.71	63.60	67.74

As shown in Table 4.1, both Qwen3-1.7B and Qwen3-4B consistently outperform comparable models of similar size across all evaluated benchmarks. In particular, Qwen3-1.7B significantly surpasses Qwen2.5-1.5B in general reasoning, mathematical, and coding tasks, demonstrating the effectiveness of architectural and training improvements despite a similar parameter scale.

Furthermore, Qwen3-4B achieves performance close to or exceeding larger models such as Qwen2.5-7B, especially in reasoning-intensive benchmarks such as GSM8K and MATH. This indicates that the Qwen3 architecture is highly efficient, enabling smaller models to retain strong reasoning capabilities typically associated with larger models.

Overall, these results highlight that Qwen3 models provide a strong balance between performance and efficiency. This makes them particularly suitable for downstream tasks such as authorship verification, where fast inference and limited computational resources are important constraints.

DeepSeek-V3.2

DeepSeek’s DeepSeek models represent a family of large language models based on the decoder-only architecture, similar to the Qwen series. These models introduce advanced capabilities in reasoning and tool usage, making them competitive on a range of complex tasks [7].

A key characteristic of DeepSeek models is the use of the *Mixture-of-Experts (MoE)* architecture. In this paradigm, the total parameter count can reach hundreds of billions; however, only a subset of parameters (referred to as experts) is activated for each token during inference. This design enables high model capacity while maintaining computational efficiency [7].

In this work, the DeepSeek-V3.2 model is selected as a representative of large-scale models, with a total parameter count of approximately 685 billion. Due to the MoE mechanism, only a fraction of these parameters (approximately 30–40 billion) is active during a single forward pass, resulting in feasible inference speed despite the model’s overall scale.

GPT-OSS

GPT-OSS refers to a family of open-weight language models developed by OpenAI. Rather than representing a single model, GPT-OSS encompasses a broader category of architectures, training paradigms, and publicly available implementations inspired by the original GPT family [40].

For the purposes of this work, the *GPT-OSS-120B* model is utilized as a high-capacity representative of this family. Together with the DeepSeek models, it serves as an example of large-scale language models evaluated in this study. Compared to other models, GPT-OSS-120B is expected to exhibit slower inference speed due to its size, but potentially improved performance¹.

4.3.2 Setup and Evaluation

Model inference was conducted using two different approaches, depending on model size and computational requirements. The smaller models, *Qwen3-1.7B* and *Qwen3-4B*, were executed locally on a workstation (see Appendix C.1). In contrast, larger models, such as DeepSeek and GPT-OSS, were deployed via an AIaaS (Artificial Intelligence as a Service) infrastructure provided by MetaCentrum, due to their significantly higher computational demands.

All models were evaluated under a unified configuration to ensure comparability:

- `max_length` = 512: Maximum length of the input text
- `temperature` = 0.01: Very low temperature to ensure deterministic and reproducible outputs
- `max_new_tokens` = 1024: Increased limit to accommodate extended reasoning in “thinking” models
- `top_p` = 0.9: Standard nucleus sampling threshold used in most practical applications

In addition to these parameters, the *thinking* mode was enabled or disabled depending on the specific capabilities and configuration of each model.

The experiments were conducted on a reduced dataset comprising 5,000 pairs due to computational constraints and token rate limitations. Nevertheless, this sample size is sufficient to provide a statistically meaningful evaluation and to demonstrate the impact of semantic properties in the dataset construction process.

Prompt

A generic, concise, and explicit instruction was used as the prompt for all evaluated models. The same template was consistently applied across all experiments to ensure comparability.

¹In the context of large language models, performance refers to the model’s ability to generate accurate responses and correctly interpret user intent.

The placeholders $\{text1\}$ and $\{text2\}$ were dynamically replaced with the corresponding text pairs from the dataset.

Text1:
{text1}

Text2:
{text2}

Same author? Answer only YES or NO and nothing more!

This prompt design enforces a strictly binary response format, simplifying the evaluation process and reducing ambiguity in model outputs.

Evaluation

A total of five experiments were conducted using different models. In the case of GPT-OSS, two configurations were evaluated: with and without the *thinking* mode enabled. The results of all experiments are summarised in Table 4.2. The metric we selected, unlike in other solutions, is MCC (explained in Section 2.2.3), which, due to its generally lower values, reflects the discriminative power more effectively than the standard ROC-AUC.

Table 4.2: Comparison of model performance using MCC on the custom dataset and PAN21 benchmark. Best results are highlighted in bold.

	Model	Params	Thinking	Own (MCC)	PAN21 (MCC)
Exp 1	Qwen3	1.7B	No	0.1194	0.0066
Exp 2	Qwen3	4B	No	0.0504	0.0792
Exp 3	DeepSeek	685B ²	Yes	0.2171	0.4296
Exp 4	GPT-OSS	120B	No	0.0588	0.0849
Exp 5	GPT-OSS	120B	Yes	0.0303	0.0968

As shown in Table 4.2, most evaluated models achieve MCC values close to zero on the constructed dataset, indicating performance near random guessing. This suggests that the task, as formulated in this work, is highly challenging and cannot be reliably solved by relying on superficial patterns or semantic similarity alone.

In contrast, higher MCC values are observed in the PAN21 dataset (except for the Qwen3-1.7B model), indicating improved recognition performance in this setting. The most notable performance is achieved by the DeepSeek model, which significantly outperforms the other models and demonstrates a stronger ability to capture author-specific features.

For the Qwen models, additional experiments were conducted with the thinking mode enabled. The results indicate a slight decrease in performance compared to the standard configuration; however, the observed difference was not statistically significant.

4.3.3 Conclusion

Overall, based on the results in Table 4.2, the semantically diverse dataset constructed in this work proves to be challenging for most evaluated models. The low MCC values

²Number of parameters during forward pass is lower. See Section 4.3.1.

indicate that models struggle when semantic similarity is not a reliable signal for authorship verification. This supports the hypothesis that many models rely heavily on semantic cues. When these cues are weakened through semantically distant positive pairs and semantically similar negative pairs, performance degrades significantly.

Interestingly, the GPT-OSS model achieves even lower performance than the smallest Qwen3-1.7B model, despite its substantially larger size. This suggests that increased model capacity alone does not guarantee improved performance in this setting, particularly when the task requires disentangling stylistic features from semantic content.

In contrast, the DeepSeek model achieves the best results, especially on the PAN21 dataset, where semantic information remains more informative. This further highlights that current models perform better when semantic similarity aligns with authorship, but struggle in more challenging, semantically diverse scenarios.

An interesting observation emerges from comparing two variants of the same Qwen3 architecture with different parameter sizes, namely 1.7B and 4B. Despite sharing the same architectural design, the smaller Qwen3-1.7B model significantly outperforms the larger 4B variant. This suggests that increased model size does not necessarily translate into better performance for the given task, and that smaller models may generalise more effectively under the specific experimental conditions.

Chapter 5

Encoder-Based Authorship Verification

This chapter introduces an encoder-based approach to authorship verification built around two transformer encoder streams that share weights and communicate during processing. Instead of encoding each text in isolation, as in the standard Siamese setup, the proposed architecture introduces co-attention layers at intermediate encoder depths, allowing each text to attend to the other text’s representation before the final pooling step. This design preserves distinct per-text representations while still enabling fine-grained cross-document comparison and provides a natural place to attach auxiliary objectives that target stylistic rather than semantic features.

5.1 Motivation

The most straightforward approach to Authorship Verification with pre-trained transformers is a Siamese network: encode each text independently with a shared BERT encoder, pool the resulting representations, and feed the pair $[u, v, |u - v|, u \odot v]$ into a classification head. This is efficient and well-studied, but it has a fundamental limitation - the two encodings are computed in complete isolation.

5.1.1 Cross-Document Attention

The model processes the concatenated input without explicitly modelling structured interactions between the two texts. As a result, the representations of each text are learned largely independently, with only implicit interaction through the shared context.

To address this limitation, cross-document attention is introduced at intermediate layers of the model, as can be seen in Figure 5.1. The goal is to enable direct interaction between the representations of the two texts, allowing each text to inform the contextual encoding of the other before the final pooling step.

At each injection point, token representations from text A attend to the representations of text B, and vice versa. This interaction is implemented using a standard multi-head attention mechanism, followed by residual connections and layer normalization. Through this process, the model can explicitly compare and align features across the two texts. Such cross-document attention allows the model to capture more nuanced relationships, such as similarities or discrepancies in syntactic structures, lexical choices, and stylistic patterns.

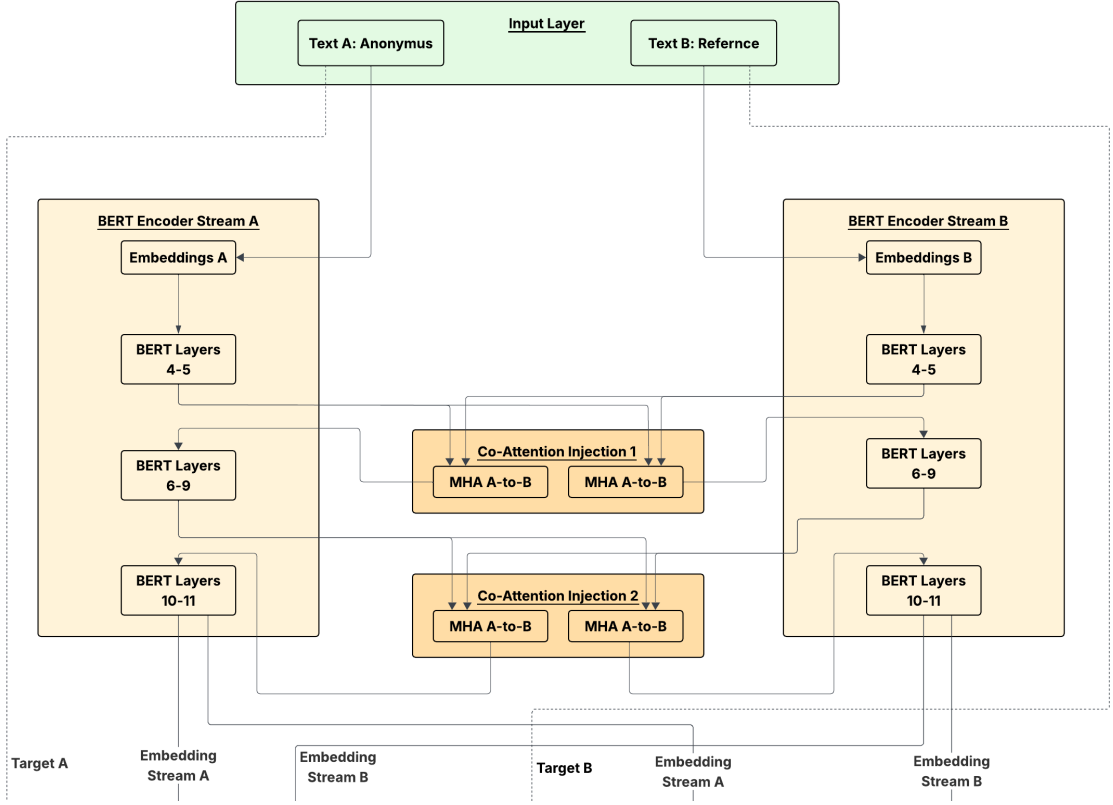


Figure 5.1: Encoder part of the proposed model based on BERT backbone, for full size please see the Appendix [A.1](#)

Consequently, the resulting representations are more informative and better suited for distinguishing whether the texts were written by the same author.

5.1.2 Style Reconstruction as a Representation Regulariser

A known limitation of encoder-based models, such as BERT, when fine-tuned on semantic or topic-oriented tasks, is their tendency to prioritize high-level semantic information while discarding low-level stylistic features. However, in the context of authorship verification, these stylistic cues constitute the primary signal of interest.

To mitigate this issue, an auxiliary objective in the form of a style reconstruction loss is introduced. In this approach, the encoder’s hidden representations serve as cross-attention context for a frozen BERT-based decoder, which reconstructs the original input text. This reconstruction process forces the encoder to preserve sufficient information about the text’s surface form, including its stylistic and lexical characteristics.

The auxiliary loss is combined with the main classification objective and weighted by a coefficient λ as is shown in Figure [5.2](#). By jointly optimising both objectives, the encoder is encouraged to retain stylistically relevant features that might otherwise be suppressed during purely discriminative fine-tuning. This results in richer representations that are better aligned with the requirements of the authorship verification task.

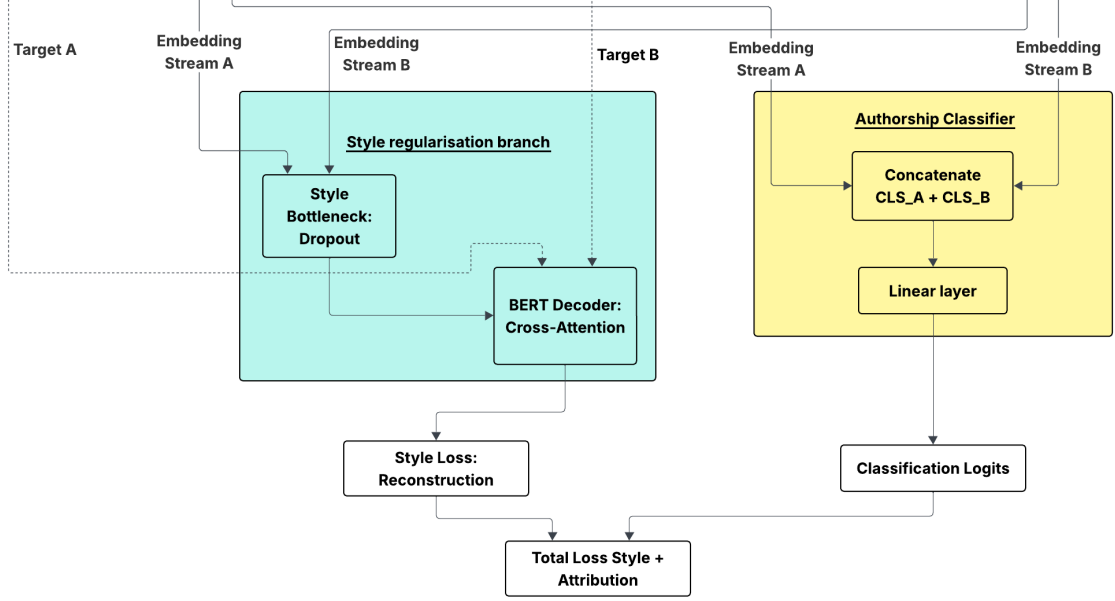


Figure 5.2: Decoder and classification parts of the proposed model, for full size please see the Appendix A.1

5.2 Model Architecture

The proposed model extends the principles of cross-document attention and style regularisation, introduced in Section 5.1, within an Encoder–Decoder framework. It builds upon transformer-based approaches discussed in Section 3.4. The architecture consists of three tightly coupled components: a shared transformer encoder, a partially frozen transformer decoder acting as a style regularizer, and a pairwise classification head. A schematic overview of the architecture is presented in Appendix A.1, while the complete forward-pass data flow is illustrated in Figure 5.3.

5.2.1 Input Layer

As shown in the top section of Figure 5.1, the model receives two independent texts: Text A and Text B. Both texts are tokenised independently using the *WordPiece* tokeniser, with sequences truncated or padded to a maximum length of 512 tokens, as defined in the configuration.

This process produces two sequences of token identifiers, denoted as ids_a and ids_b , along with their corresponding attention masks, $mask_a$ and $mask_b$. These serve as inputs to the two logical encoder streams.

5.2.2 Shared Encoder with Co-Attention Injections

Both text streams are processed using a single shared encoder backbone. Although Figure 5.1 depicts two conceptual streams (Stream A and Stream B), these do not correspond to separate model instances. Instead, the input sequences are concatenated along the batch dimension and processed jointly in a single forward pass through the encoder, as illustrated in Figure 5.3. After encoding, the combined output is split back into the respective repre-

sentations $\mathbf{x}$ (Text A) and $\mathbf{y}$ (Text B). This batched processing strategy avoids redundant computation while preserving logical separation between the two inputs.

The key architectural extension is the introduction of a bidirectional co-attention mechanism (as described in Section 5.1.1), represented by the *Co-Attention Injection* blocks in Figure 5.1. These cross-attention modules are inserted into selected encoder layers via forward pre-hooks, enabling the modification of intermediate representations without altering the underlying BERT implementation.

At each injection point, the hidden representations are split into $\mathbf{x}$ and $\mathbf{y}$ and passed through a `CoAttentionLayer`. This layer performs two complementary multi-head attention operations: $MHA^1 A \rightarrow B$, where Text A attends to Text B, and $MHA B \rightarrow A$, where Text B attends to Text A. Each operation is followed by residual connections and layer normalisation.

The outputs of the co-attention layer are incorporated using a learnable scalar gate, as defined in Equation:

$$\mathbf{x}' = \mathbf{x} + \sigma(g) \cdot (\tilde{\mathbf{x}} - \mathbf{x}), \quad \mathbf{y}' = \mathbf{y} + \sigma(g) \cdot (\tilde{\mathbf{y}} - \mathbf{y}) \quad (5.1)$$

where $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ denote the co-attended representations, and $\sigma(g)$ is a sigmoid-activated scalar parameter.

The gate parameter is initialised to -1.5 , corresponding to $\sigma(g) \approx 0.18$, which ensures that cross-document interactions are initially weak. This allows the model to first learn strong independent representations for each text before gradually incorporating cross-document information. Such a design prevents the co-attention mechanism from acting as a shortcut during early training stages.

5.2.3 The Style Reconstruction Branch

After the final encoder layer, the hidden representations of both input streams are passed to the Style Reconstruction Branch, illustrated on the left side of Figure 5.2. The representation of Text A is denoted as $\mathbf{x}$, while the representation of Text B is denoted as $\mathbf{y}$. The branch consists of four main components: a style dropout, a detached context interface, a partially frozen decoder, and a reconstruction loss.

The style dropout is implemented as a dropout layer with probability p . During each forward pass, dropout randomly masks parts of the hidden representations before they enter the reconstruction branch. Since the output is detached in the next step, this dropout does not directly regularise the encoder. Instead, it regularises the reconstruction branch by forcing the language modelling head to work with slightly perturbed encoder representations, reducing its dependence on individual feature dimensions.

The next step is the *detached context* interface. Here, the hidden representations are detached from the computational graph before being passed to the decoder. This design follows the semantics-agnostic motivation introduced in Section 5.1. A token-level reconstruction objective naturally encourages the model to preserve lexical and topical information, because this information helps reconstruct the original text. However, this is not desirable for stylometric verification, where the model should focus on writing style rather than content. If the reconstruction gradients were allowed to update the encoder, they could push the encoder toward content-preserving representations. By detaching the context, this gradient path is blocked, as shown in Figure 5.3. As a result, the encoder is

¹MHA stands for Multi-Head Attention. See section 3.1.5.

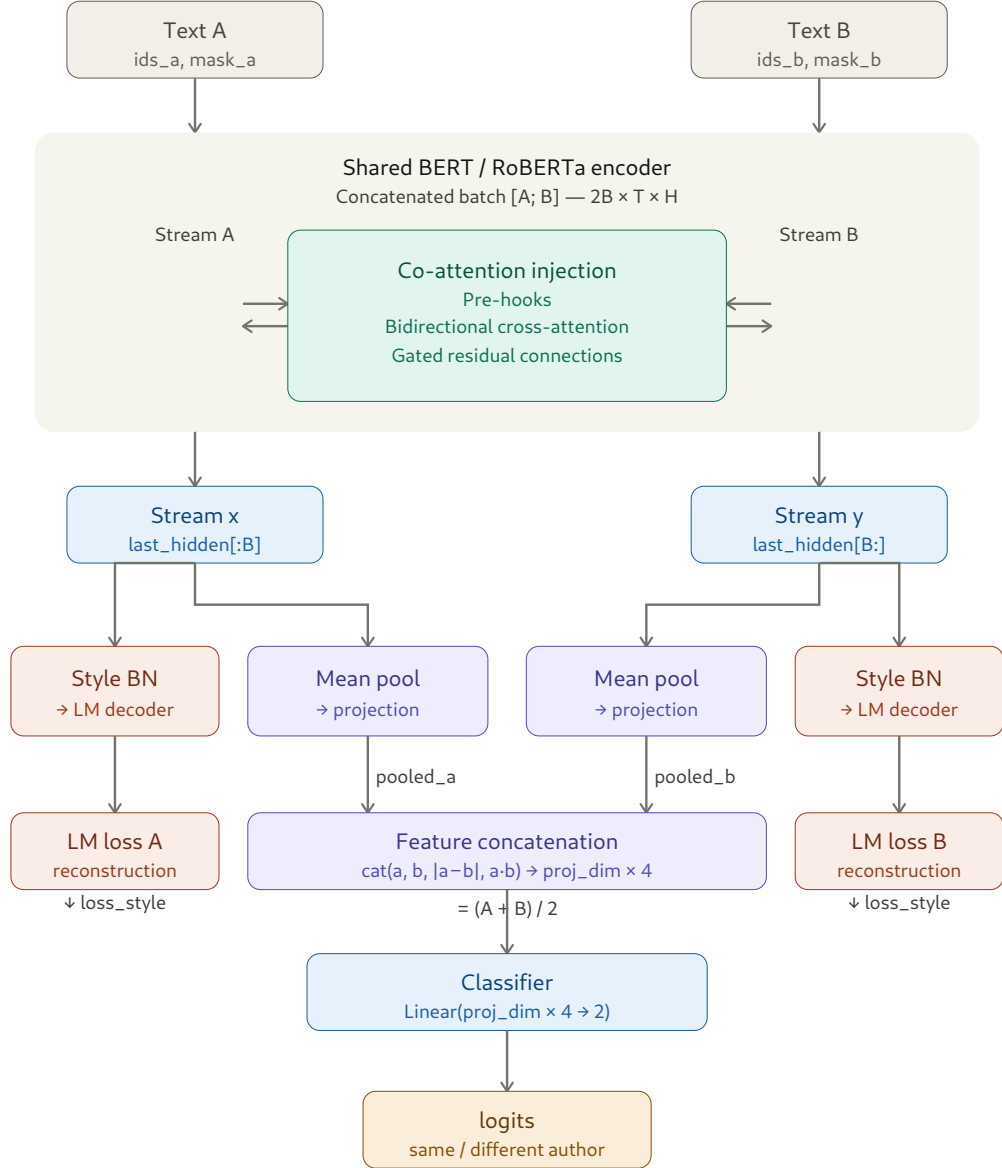


Figure 5.3: Visualisation of the proposed architecture using a BERT or RoBERTa encoder.

optimised only by the classification objective and, where applicable, the contrastive objective, while the reconstruction branch acts only as a probe of the information retained in the encoder representations.

The central component of the branch is a transformer decoder. During training, all transformer layers of the decoder are frozen, and only the language modelling head is trainable. This allows the head to adapt to the encoder representations without changing the decoder itself. At the same time, freezing the decoder limits its ability to learn shortcuts that would make the reconstruction loss less informative. The decoder is used only during training and is discarded at inference time, leaving only the encoder and classification head for prediction.

The decoder tries to reconstruct the original input tokens from the encoder representations. The reconstruction loss is computed using causal language modelling cross-entropy.

Since both input streams are reconstructed, the final style reconstruction loss is defined as the average of the two stream-specific losses:

$$\mathcal{L}_{\text{style}} = \frac{\mathcal{L}_{\text{style_A}} + \mathcal{L}_{\text{style_B}}}{2}. \quad (5.2)$$

Although $\mathcal{L}_{\text{style}}$ does not directly update the encoder, it is still useful during training. First, it trains the language modelling head to reconstruct tokens from the current encoder representations. The weighting coefficient λ in the combined objective $\mathcal{L}_{\text{cls}} + \lambda \cdot \mathcal{L}_{\text{style}}$, described in Section 5.2.5, controls how strongly this branch is trained during different training phases. Second, the reconstruction loss serves as a diagnostic signal. If $\mathcal{L}_{\text{style}}$ increases during fine-tuning, it suggests that the encoder is discarding more surface-level lexical information, which supports the semantics-agnostic goal of the architecture.

5.2.4 Authorship Classifier

The classifier branch, shown as the yellow *Authorship Classifier* block in Figure 5.2, operates on the final encoder outputs $\mathbf{x}$ and $\mathbf{y}$ produced by the two shared encoder streams. To obtain fixed-size document-level representations, masked mean pooling is applied over all non-padding token positions. This operation computes a weighted average of valid token embeddings and has been found to produce robust representations, particularly for longer documents.

The pooled representations are then passed through a shared projection head. This projection head consists of a linear layer that reduces each document representation to a 256-dimensional bottleneck vector, followed by a GELU activation function. GELU was selected to maintain consistency with the activation functions used within the encoder backbone (see Section 3.1.6). A dropout probability of 0.2 is applied after the projection stage to improve regularisation and reduce overfitting.

The resulting projected vectors, denoted as $\mathbf{u}$ for Text A and $\mathbf{v}$ for Text B, are combined into a four-part interaction vector designed to capture complementary aspects of their relationship. The final interaction vector is formed by concatenating the two projected vectors :

$$\mathbf{z} = [\mathbf{u}; \mathbf{v}; |\mathbf{u} - \mathbf{v}|; \mathbf{u} \odot \mathbf{v}], \quad (5.3)$$

their absolute element-wise difference, and their Hadamard product, which is defined:

$$\mathbf{u} \odot \mathbf{v} = (u_1 v_1, u_2 v_2, \dots, u_d v_d). \quad (5.4)$$

This formulation produces a 1024-dimensional feature vector for classification. The absolute difference term $|\mathbf{u} - \mathbf{v}|$ captures the magnitude of feature-wise discrepancies between the two representations, while the Hadamard product $\mathbf{u} \odot \mathbf{v}$ captures feature co-activation patterns. Together, these interaction features provide the classifier with richer information about both similarity and divergence between the two texts, improving its ability to distinguish whether they originate from the same author.

5.2.5 Total Training Objective

As illustrated in Figure 5.2, the final training objective combines the classification loss and the style reconstruction loss into a single optimisation criterion. The total loss is defined as :

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda \cdot \mathcal{L}_{\text{style}}, \quad (5.5)$$

where $\mathcal{L}_{\text{cls}}$ denotes one of the classification loss functions described in Section 3.1.7, and $\mathcal{L}_{\text{style}}$ represents the average reconstruction loss produced by the decoder branch.

The weighting coefficient λ controls the relative influence of the auxiliary style reconstruction objective during training. Its value is governed by a two-phase schedule specified in the configuration file. In the initial phase, a lower value of λ is used to allow the model to first stabilise the main classification objective. In the later phase, the contribution of the style reconstruction loss is increased, encouraging the encoder to preserve richer stylistic information while maintaining discriminative performance for authorship verification.

This joint optimisation strategy balances the primary goal of accurate classification with the auxiliary objective of retaining stylistically relevant features, leading to more robust and informative document representations.

5.3 Training

Two encoder backbones were selected for training: BERT and the larger RoBERTa model. Due to their differing computational requirements, training was carried out on two separate machines: BERT-based models were trained on the workstation described in Appendix C.2, while RoBERTa-based models were trained on the infrastructure described in Appendix C.3.

Although each backbone was evaluated under multiple experimental configurations, several training settings were shared across all runs. Gradient accumulation (described in Section 6.2) was applied with an accumulation factor of 2 to simulate a larger effective batch size while remaining within GPU memory constraints. Each model was trained for a maximum of 12 epochs, with early stopping applied using a patience of 3 epochs. The best checkpoint was selected according to ROC-AUC, as described in Section 2.2.2.

As in the decoder-based approach, the maximum input length was limited to 512 tokens. This setting balances computational efficiency with sufficient textual coverage while remaining compatible with GPU memory limitations.

For optimisation, a non-fused version of the *AdamW* optimiser was used. This choice was made primarily to support multiple parameter groups with different learning rates. The weight decay coefficient was set to 0.01, following standard regularisation practice.

A learning rate scheduler combining linear warm-up and linear decay was employed. The warm-up phase spans the first 6% of total training steps, during which the learning rate increases gradually from zero to its target value. After warm-up, the learning rate decreases linearly to zero over the remaining steps, ensuring stable optimisation in early training and smooth convergence later.

An important aspect of training is the use of discriminative learning rates. Encoder parameters are updated with lower learning rates to preserve pre-trained linguistic knowledge, while newly initialised layers are trained more aggressively. For BERT, the encoder learning rate was set to 2×10^{-5} , while for RoBERTa it was reduced to 1×10^{-5} . Newly introduced components, including co-attention layers, projection heads, and classifier layers, were trained with a higher learning rate of 1×10^{-4} for both backbones.

Three loss functions described in Section 3.1.7 were evaluated during training. For focal loss, the parameter α was set to 0.5 for both backbone configurations, reflecting the balanced nature of the constructed dataset (see Section 3.1.7), while γ was fixed at 2.0, following the canonical setting proposed in the original Focal Loss paper [32].

For Supervised Contrastive Loss, the hyperparameters were selected separately for each backbone. For the smaller BERT backbone, the temperature was set to $\tau = 0.7$ and the contrastive weight to 0.5. At this value, similarities are scaled strongly, producing a sharper contrastive signal that helps compensate for the lower representational capacity of the smaller model. For RoBERTa, which uses a larger projection dimension of 512, a lower temperature of $\tau = 0.15$ was chosen to soften the similarity distribution and improve training stability. Preliminary experiments showed that lower temperatures combined with higher contrastive weights often caused unstable optimization, motivating the selected configuration.

When Supervised Contrastive Loss is used, it is combined with the standard cross-entropy classification loss and the style reconstruction loss, as defined as follows:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + w_{sc} \cdot \mathcal{L}_{\text{SupCon}} + \lambda \cdot \mathcal{L}_{\text{style}}. \quad (5.6)$$

The main differences between the training configurations of both encoder backbones are summarized in Table 5.1.

Table 5.1: Hyperparameters between the `bert-base-uncased` and `roberta-large` CoAttentionBert configurations. All other hyperparameters are identical across both backbones.

Hyperparameter	BERT	RoBERTa
<i>Backbone</i>		
Base model	<code>bert-base-uncased</code>	<code>roberta-large</code>
Hidden size	768	1 024
Trained on	See Appendix C.2	See Appendix C.3
<i>Architecture</i>		
Injection layers	6, 9	8, 16
Attention heads	8	16
Projection dim	256	512
<i>Optimiser</i>		
Encoder LR	2×10^{-5}	1×10^{-5}
New LR	1×10^{-4}	1×10^{-4}
<i>Training</i>		
Batch size	8	16
Grad. accum. steps	2	2
Effective batch size	16	32
<i>Loss</i>		
Focal α	0.5	0.5
SupCon temperature τ	0.07	0.15
SupCon weight	0.5	0.25

Two phase training

A two-phase optimisation strategy was adopted. During *Phase 1*, covering the first 3 epochs, the encoder backbone remains fully frozen. Only the newly introduced components are updated: the co-attention layers, learnable gating parameters, projection head, classifier,

and decoder language modelling head. The decoder body itself remains frozen throughout the entire training process.

In this initial phase, the style reconstruction loss is assigned a relatively high weight of $\lambda = 0.3$ in Equation (5.5). The purpose of this stage is to allow the newly added modules to stabilise before perturbing the pre-trained encoder representations.

During *Phase 2*, spanning epochs 4 through 12, the encoder is unfrozen, and all trainable parameters are optimised jointly. At this stage, the weight of the style reconstruction loss is reduced to $\lambda = 0.05$, making the classification objective dominant. This allows the encoder to gradually adapt to the task while minimising catastrophic forgetting.

5.3.1 BERT training

We train with a batch size of 8 and gradient accumulation over 2 steps, yielding an effective batch size of 16. Cross-attention is injected at encoder layers 6 and 9. An earlier configuration was additionally injected at layer 3, but this produced unstable training dynamics. The choice of layers 6 and 9 is motivated by the 12-layer depth of BERT-base: probing studies Tenney et al., 2019; Jawahar et al. (2019) [53] suggest that early layers encode syntactic and surface-level features, middle layers stabilize semantic and discourse-level representations, and final layers concentrate task-specific signal. Injecting cross-attention in the middle range allows each text stream to attend to the other once sufficiently abstract representations have formed, without disrupting the classifier-orientated final layers.

BERT training runs

The training curves shown in Figure 5.4a illustrate the optimization process for experiments based on the BERT backbone over 12 epochs. Figure 5.4b presents the corresponding validation ROC-AUC, which serves as the primary criterion for model selection. The model trained with cross-entropy loss was terminated early in 4th epoch due to stagnation in validation performance, as indicated by the plateau in Figure 5.4b.

During *Phase 1*, all three configurations exhibit a rapid decrease in training loss. The cross-entropy and supervised contrastive loss curves decrease smoothly from approximately 2.1 and 2.4, respectively. In contrast, the focal loss curve—displayed on a separate scale due to its larger absolute magnitude—decreases from 6.96 to 5.95.

At the transition from Phase 1 to Phase 2, a noticeable drop in loss is observed across all configurations. This change coincides with unfreezing the encoder weights, enabling joint optimization of the full model. Following this transition, the cross-entropy loss stabilizes around 0.595, the supervised contrastive loss around 0.636, and the focal loss around 1.148, indicating successful adaptation of the pre-trained encoder to the task.

As the training loss stabilised, the rate of improvement in ROC-AUC gradually decreased, as illustrated in Figure 5.4b. All configurations converge to a plateau, with the cross-entropy model reaching 0.9231, the supervised contrastive variant achieving 0.9269, and the focal loss model attaining 0.9229.

Figure 5.5 illustrates the evolution of the sigmoid gate activations across training epochs. All gates increase monotonically from their initialisation baseline of $\sigma(-1.5) \approx 0.182$, indicating that the model progressively learns to incorporate cross-document information in all configurations.

Among the evaluated loss functions, the supervised contrastive (SupCon) setup exhibits the most pronounced increase in gate activation, rising from approximately 0.201 to 0.245, with the steepest growth occurring already during Phase 1. The cross-entropy configuration reaches a moderate activation level of 0.214 before early stopping is triggered. In contrast,

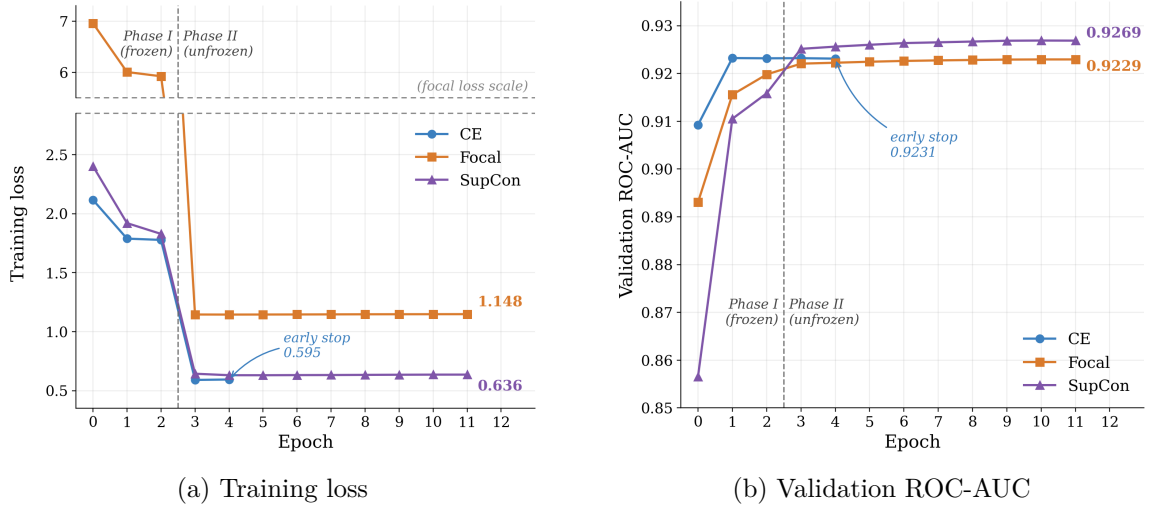


Figure 5.4: Training process of co-attention BERT model trained on 3 different loss functions: Cross-entropy, Focal loss and Supervised Contrastive Loss.

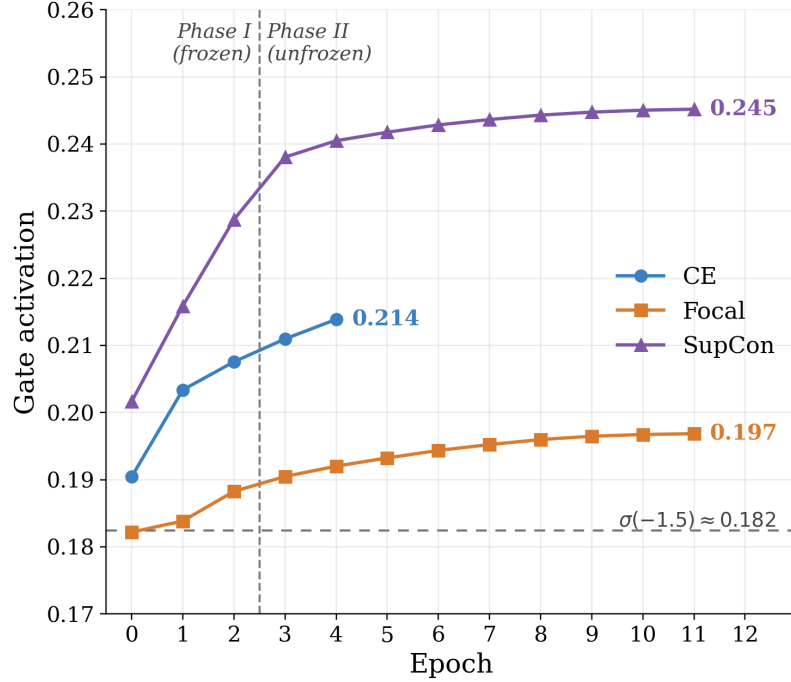


Figure 5.5: Sigmoid gate activation at co-attention BERT layer 6, which is initialised with a bias of -1.5 (corresponding $\sigma(-1.5) \approx 0.182$, indicated by the dashed reference line).

the focal loss configuration remains closest to its initial value, ending at approximately 0.197.

Interestingly, this behaviour contrasts with the fact that focal loss achieves the lowest training loss. This suggests that its emphasis on hard examples primarily benefits the classifier head, while contributing less to the development of cross-document interaction mechanisms within the model.

5.3.2 RoBERTa training

The RoBERTa-based Co-Attention model was trained with a batch size of 16 and gradient accumulation over 2 steps, resulting in an effective batch size of 32. Following the same design principles as in the BERT-based setup (see Section 5.3.1), cross-document attention was injected at encoder layers 8 and 16.

Additional experiments with three injection points were conducted to account for the increased depth of the RoBERTa model; however, no significant performance improvement was observed. Therefore, the two-injection configuration was retained for efficiency.

Due to the larger size of the RoBERTa backbone, the learning rate was reduced to 1×10^{-5} , i.e., half of the value used for BERT. Similarly, the hyperparameters of the supervised contrastive loss were scaled down to maintain training stability and prevent overly strong gradient signals in the higher-capacity model.

During training, particularly in Phase 2, the value of the style reconstruction loss increased significantly. This behaviour is expected due to the trade-off between the classification objective and the auxiliary style loss: as the weight λ is reduced and the model focuses more on classification, the reconstruction quality may deteriorate. However, the magnitude of the style loss occasionally exceeded the numerical limits of the underlying data type, leading to instability. To address this issue, loss clipping was applied during training to prevent overflow and ensure stable optimisation.

RoBERTa training runs

The training dynamics of the Co-Attention model with a RoBERTa backbone are illustrated in Figure 5.6. At the end of Phase 1 (epoch 2), training losses remain relatively high—cross-entropy (CE) and supervised contrastive loss (SupCon) around ≈ 2.2 , and focal loss around ≈ 5.9 —reflecting the limited optimization capacity when the encoder is frozen (epochs 0–1 are omitted for clarity). Following the transition to Phase 2 at epoch 3, all configurations exhibit a sharp decrease in loss. Focal loss stabilizes quickly at approximately ≈ 0.36 and remains nearly constant throughout Phase 2. In contrast, CE decreases more gradually, reaching ≈ 0.51 by epoch 11, while SupCon converges to a higher plateau at approximately ≈ 0.65 .

The corresponding ROC-AUC curves (Figure 5.6b) reveal a delayed improvement pattern for the CE and SupCon configurations. After unfreezing, both runs initially exhibit a plateau in validation performance before a sharp increase. Specifically, CE remains at approximately ≈ 0.941 for several epochs before rising to 0.975 at epoch 8, while SupCon improves earlier, reaching ≈ 0.965 at epoch 5. This behaviour suggests a gradual adaptation of the pre-trained encoder, which requires several epochs to adjust its representations to the authorship verification task.

In contrast, the focal loss configuration does not exhibit this delay. Its ROC-AUC increases immediately after unfreezing, reaching approximately ≈ 0.974 at epoch 3 and stabilizing around 0.9772. The final ROC-AUC values are 0.9779 for CE, 0.9772 for focal loss, and 0.9652 for SupCon, indicating that CE and focal loss achieve comparable performance, while SupCon lags behind in this setting.

Figure 5.7 presents the evolution of the sigmoid gate activations at co-attention layer 9. In all configurations, the gate values increase monotonically from the initialisation baseline of $\sigma(-1.5) \approx 0.182$, indicating that cross-document attention is progressively utilised during training.

Among the evaluated loss functions, the supervised contrastive (SupCon) configuration exhibits the strongest gate activation, increasing from 0.211 to 0.270, which is consistent

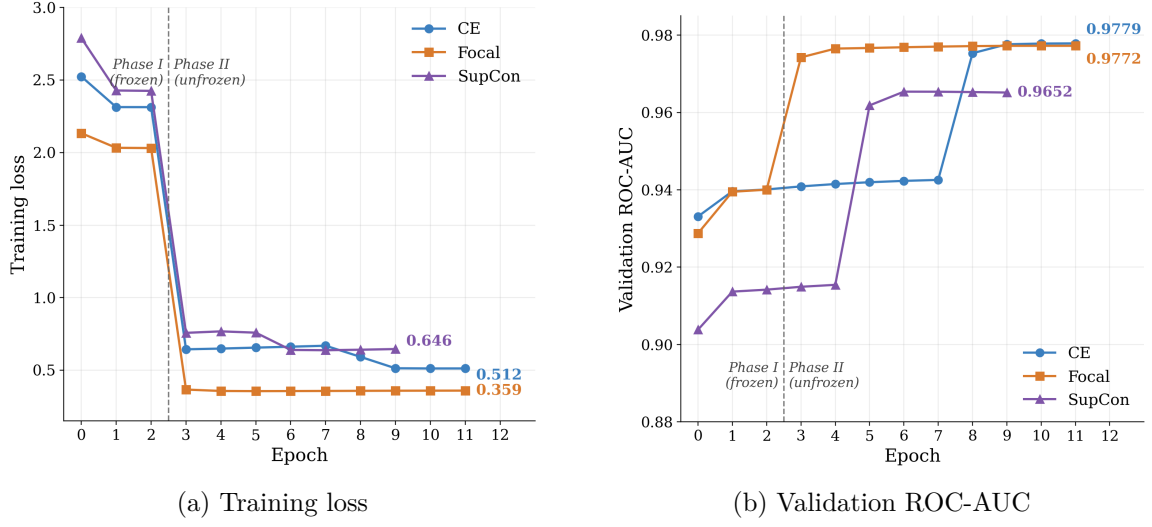


Figure 5.6: Training process of Co-Attention RoBERTa model trained on 3 different loss functions: Cross-entropy, Focal loss and Supervised Contrastive Loss.

with the behaviour observed for the BERT backbone. In contrast, the focal loss configuration remains the most conservative, with gate values increasing only marginally and reaching 0.207 at the end of training. This suggests that focal loss relies less on cross-document interaction and more on the classifier’s ability to discriminate difficult examples.

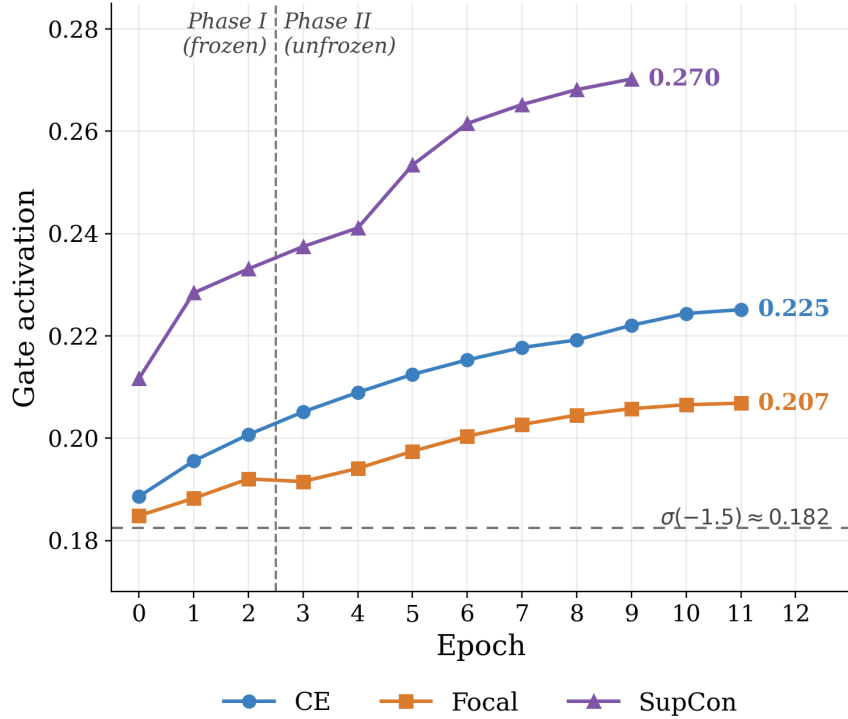


Figure 5.7: Sigmoid gate activation at co-attention RoBERTa layer 8, which is initialised with a bias of -1.5 (corresponding $\sigma(-1.5) \approx 0.182$, indicated by the dashed reference line).

5.4 Evaluation

The evaluation results obtained from the trained models are presented in Table 5.2 for the custom dataset and Table 5.3 for the PAN21 benchmark. On the custom dataset, the co-attention model based on the BERT backbone achieves ROC-AUC values in the range of 0.91–0.92 and MCC values between 0.66–0.68, while the RoBERTa-based variant significantly outperforms it, reaching 0.96–0.98 ROC-AUC and 0.79–0.84 MCC values. This confirms the benefit of increased model capacity and richer contextual representations.

Within the RoBERTa-large variants, the cross-entropy loss achieves the highest overall performance, with focal loss yielding comparable results across all metrics. The SupCon variant trails both by a moderate margin — an AUC of 0.9647 compared to 0.9767 for CE — which may partly reflect the numerical instability encountered during training rather than a fundamental limitation of the loss function. Among BERT-based models, differences between loss functions are small and largely within noise, with SupCon achieving the highest AUC (0.9209) but lower F1 and MCC (0.6693), consistent with a slight threshold miscalibration.

Table 5.2: Evaluation results on the custom dataset. Best result per column is in **bold**.

Model	AUC	EER	F1 ^a	MCC ^a	Bal. Acc ^a
BERT (CE)	0.9149	0.1653	0.8345	0.6679	0.8340
BERT (Focal)	0.9158	0.1626	0.8375	0.6733	0.8367
BERT (SupCon)	0.9209	0.1589	0.8265	0.6693	0.8335
RoBERTa-large (Focal)	0.9760	0.0812	0.9085	0.8240	0.9113
RoBERTa-large (CE)	0.9767	0.0785	0.9150	0.8351	0.9171
RoBERTa-large (SupCon)	0.9647	0.1027	0.8982	0.7938	0.8967

^a Reported at the best-F1 threshold.

On the PAN21 benchmark (Table 5.3), a consistent drop in performance is observed across all models, reflecting the increased difficulty of this out-of-domain setting with unseen authors and domain shift. BERT-based models achieve ROC-AUC values between 0.83–0.85 and MCC values between 0.48–0.51, while RoBERTa-based models maintain stronger ROC-AUC in the range of 0.91–0.93 and MCC values between 0.65–0.71.

Table 5.3: Evaluation results on the PAN21 dataset. Best result per column is in **bold**.

Model	AUC	EER	F1 ^a	MCC ^a	Bal. Acc ^a
BERT (CE)	0.8486	0.2348	0.7786	0.5122	0.7406
BERT (Focal)	0.8483	0.2318	0.7822	0.5221	0.7472
BERT (SupCon)	0.8349	0.2467	0.7672	0.4827	0.7246
RoBERTa-large (Focal)	0.9263	0.1513	0.8484	0.6969	0.8484
RoBERTa-large (CE)	0.9335	0.1435	0.8598	0.7083	0.8519
RoBERTa-large (SupCon)	0.9182	0.1605	0.8380	0.6563	0.8199

^a Reported at the best-F1 threshold.

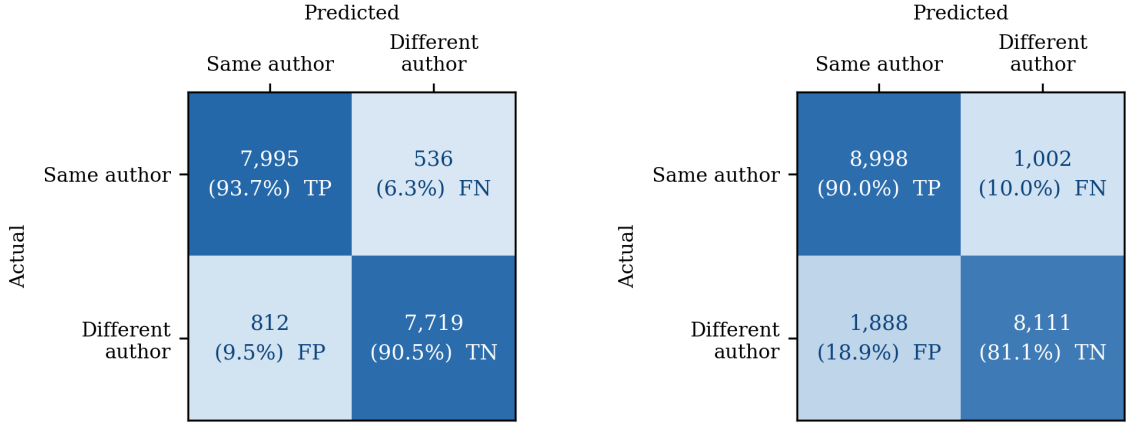
The generalisation gap between in-domain and out-of-domain performance is quantified in Table 5.4 for the best-performing configuration (RoBERTa-large with cross-entropy). While performance degradation is expected, the drop in MCC (−0.1268) and increase in

Table 5.4: Generalisation gap for Co-Attention model with RoBERTa-large (CE) between the in-distribution custom test set and the PAN21 out-of-domain benchmark.

Metric	Custom	PAN21	Drop
AUC	0.9767	0.9335	−0.0432
EER	0.0785	0.1435	+0.0650
F1	0.9150	0.8598	−0.0552
MCC	0.8351	0.7083	−0.1268

EER (+0.0650) highlight the challenges associated with cross-domain authorship verification.

Balanced accuracy, which averages recall over both classes and is therefore insensitive to class imbalance, closely tracks F1 across all configurations. For the best-performing model, it drops from 0.9171 on the custom set to 0.8519 on PAN21, corroborating the pattern observed in MCC and confirming that the generalisation gap is not an artefact of threshold selection.



(a) Custom test set ($n = 17,062$, $\tau^* = 0.2864$).

(b) PAN21 set ($n = 19,999$, $\tau^* = 0.5327$).

Figure 5.8: Confusion matrices of the Co-Attention model at the best- F_1 threshold on the custom test set and the PAN21 held-out benchmark. Cell colours reflect row-normalised recall per actual class; raw counts and row percentages are shown in each cell.

Further insight is provided by the confusion matrices in Figure 5.8. On the custom test set ($\tau^* = 0.2864$), the model achieves a recall of 93.7% for same-author pairs and a specificity of 90.5% for different-author pairs, indicating a well-balanced decision boundary.

On the PAN21 dataset ($\tau^* = 0.5327$), performance degradation is more pronounced for negative pairs. The false positive rate increases from 9.5% to 18.9%, indicating that the model becomes more prone to predicting “same author” in unseen domains. While recall decreases only moderately (93.7% $\rightarrow$ 90.0%), specificity drops more substantially (90.5% $\rightarrow$ 81.1%), suggesting that cross-domain generalisation primarily affects the model’s ability to discriminate between different authors.

5.5 Experiment Conclusion

In this approach, we presented a series of experiments adapting transformer encoders with co-attention injections and a style regularisation mechanism. Fine-tuning both BERT and RoBERTa-large backbones on the custom verification dataset resulted in strong in-domain performance, with the best configuration achieving $\text{MCC} = 0.8351$ on the custom test set (Table 5.2) and $\text{MCC} = 0.7083$ on the held-out PAN21 benchmark (Table 5.3).

The results demonstrate that the choice of backbone architecture has a substantial impact on performance. In particular, replacing BERT with RoBERTa-large yields consistent improvements across all evaluated metrics and both datasets, confirming the benefit of higher-capacity pre-trained representations.

The proposed co-attention mechanism, which introduces cross-document interactions at intermediate encoder layers via learnable sigmoid gates, proved to be both practical and effective. This validates the architectural hypothesis introduced in Section 5.1, namely that encoder-based models can be extended to capture relational stylistic cues between paired documents without requiring full cross-encoder concatenation. Such interactions are especially important in authorship verification, where subtle stylistic dependencies between texts form the primary discriminative signal.

Regarding the loss functions, cross-entropy and focal loss achieve comparable performance across both backbones, while supervised contrastive loss consistently underperforms in this setting. Overall, the results indicate that architectural design and backbone capacity play a more critical role than the specific choice of loss function once a strong training setup is established.

Chapter 6

Decoder-Based Authorship Verification with LoRA Fine-Tuning

As described in the evaluation of the Preliminary Experiment in Section 4.3, the Qwen model demonstrated promising results in distinguishing authors on our test dataset, which motivated adapting it to the authorship verification task.

Using a decoder as the backbone for classification is relatively uncommon. Its primary advantage lies in cross-document attention without explicit architectural machinery: both texts can be concatenated into a single prompt, and the decoder processes the entire sequence in one forward pass. Due to the causal attention mask, tokens belonging to the second text can attend to all tokens of the first text, enabling implicit cross-document interaction without the need for hand-engineered components such as the co-attention layers described in Section 5.

A further motivation stems from the pretraining objective of causal language models. To accurately predict each next token, a model must implicitly learn fine-grained stylistic regularities across large corpora — including lexical preferences, syntactic habits, and discourse patterns. These are precisely the signals that authorship verification seeks to exploit. This stands in contrast to masked language models such as BERT, whose pretraining objective requires reconstructing individual tokens in isolation, providing a weaker inductive bias toward holistic style modelling.

A further advantage is model scale. Even compact decoder models such as Qwen3-1.7B contain approximately 1.7 billion parameters, compared to 110 million in BERT-base, providing substantially greater representational capacity for capturing subtle stylistic patterns.

6.1 Model Architecture

The decoder-based approach is implemented as a decoder-based pair classifier built on a decoder-only backbone. As outlined in Section 6, the key motivation is that concatenating both texts into a single sequence allows the model to achieve implicit cross-document interaction through its native self-attention mechanism, without requiring any architectural modification. The full architecture is illustrated in Figure 6.1.

6.1.1 Prompt Concatenation

During both training and inference, each text pair is formatted into a single input sequence using the following prompt template:

Determine whether the following two texts were written
by the same author.

Text A:
{text_a}

Text B:
{text_b}

Answer:

This formulation is natural for decoder-only Transformer architectures, which are designed to process task instructions directly within the input sequence. Providing an explicit task description guides the model toward the intended objective and promotes consistent behaviour across training and inference. The total input (two texts with static prompt) is limited to 1024 tokens; after reserving a fixed budget for the prompt template itself, the remaining tokens are split evenly between the two texts, ensuring both are represented proportionally while respecting the model’s context limit.

6.1.2 Backbone

Qwen3-1.7B serves as the primary backbone architecture, with Qwen3-4B evaluated as a scaling comparison. Both models are described in detail in Section 4.3.1. The primary motivation for selecting a decoder-only model over encoder-based alternatives is the alignment between the autoregressive pretraining objective and the stylometric task: next-token prediction over large corpora incentivises the model to capture fine-grained stylistic regularities — lexical preferences, syntactic patterns, and discourse rhythm — since accurately predicting the next token requires modelling *how* an author writes, not merely *what* they write about. Among decoder-only models of comparable scale, Qwen3-1.7B and Qwen3-4B were selected for their favourable performance-to-size ratio at a scale not available in other open model families.

A consequence of the autoregressive attention mask, common to all decoder-only Transformers, is that cross-document interaction is asymmetric: tokens in *Text B* can attend to all tokens in *Text A*, but tokens in *Text A* have no access to *Text B*. This is a known limitation relative to bidirectional encoder-based cross-attention.

The backbone weights are kept frozen throughout training, with adaptation performed exclusively via LoRA [15] (described in Section 3.1.6). LoRA adapters are applied to all four attention projection matrices: query, key, value, and output projections (`q_proj`, `k_proj`, `v_proj`, `o_proj`) across all transformer layers, as shown in Figure 6.1. The rank is set to $r = 16$ with scaling factor $\alpha = 32$, yielding an effective update scaling of $\frac{\alpha}{r} = 2.0$, so that the adapted weight matrix is given by Equation:

$$W' = W_0 + \frac{\alpha}{r}BA, \tag{6.1}$$

where $W_0 \in \mathbb{R}^{d \times k}$ is the frozen pre-trained weight, $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$ are the trainable low-rank matrices, and $r \ll \min(d, k)$ [15]. A dropout rate of 0.05 is applied within the LoRA layers for regularisation. The classification head is excluded from LoRA and trained in full, as it is randomly initialised and has no pre-trained weights to preserve.

The sequence is pooled by extracting the hidden state of the final non-padding token. This is the natural pooling strategy for causal language models: due to the autoregressive mask, the last token has attended to every preceding token in the sequence, making it the most contextually complete representation. Mean pooling is less appropriate here because tokens belonging to *Text A* were processed without access to *Text B*, and including them equally would incorporate representations formed without full cross-document context.

6.1.3 Classification Head

A single linear layer projects the pooled representation from the backbone hidden dimension of 2048 to two output classes, corresponding to the same-author and different-author labels. Dropout with $p = 0.1$ is applied to the pooled representation prior to the linear projection as a regularisation measure. The head is trained from random initialisation using standard cross-entropy loss.

6.2 Training

Training was conducted on the hardware described in Appendix C.1, which proved sufficient for fine-tuning Qwen3 at the 4B parameter scale. Although decoder-only transformers support considerably longer context windows than encoder models, the maximum sequence length was capped at 512 tokens per input pair. Within this budget, each of the two texts receives approximately half of the tokens remaining after the static prompt template is tokenised, yielding a roughly equal token allocation per document. This constraint was imposed both to enable a fair comparison with the encoder-based approach and to limit GPU memory consumption.

To make fine-tuning tractable given the scale of the model and dataset (see Section 4.2.2), several standard training optimisations were employed. Automatic Mixed Precision (AMP) was applied using BFloat16, which reduces memory bandwidth by representing intermediate activations in a 16-bit floating-point format rather than the standard 32-bit format, with negligible impact on numerical stability for this task. Gradient accumulation with an accumulation step of 8 was used to decouple the per-device batch size from the effective batch size: with a per-device batch size of 16, the effective batch size becomes $16 \times 8 = 128$ samples per update. Gradient check-pointing was additionally enabled, trading additional forward-pass computation for a substantial reduction in activation memory by recomputing intermediate activations during the backward pass rather than retaining them. Despite these optimisations, training remained slow relative to the encoder-based approach; accordingly, the training set was capped at 300,000 pairs.

Training employed two loss functions: cross-entropy (Section 3.1.7) and focal loss (Section 3.1.7). Supervised Contrastive loss was not applicable, as the QwenPairClassifier operates as a cross-encoder: both texts are concatenated into a single sequence prior to the forward pass, meaning the pooled representation is inherently pair-dependent rather than a standalone embedding of either text. Supervised contrastive learning requires per-sample embeddings that can be meaningfully contrasted across the batch, a condition this architecture cannot satisfy.

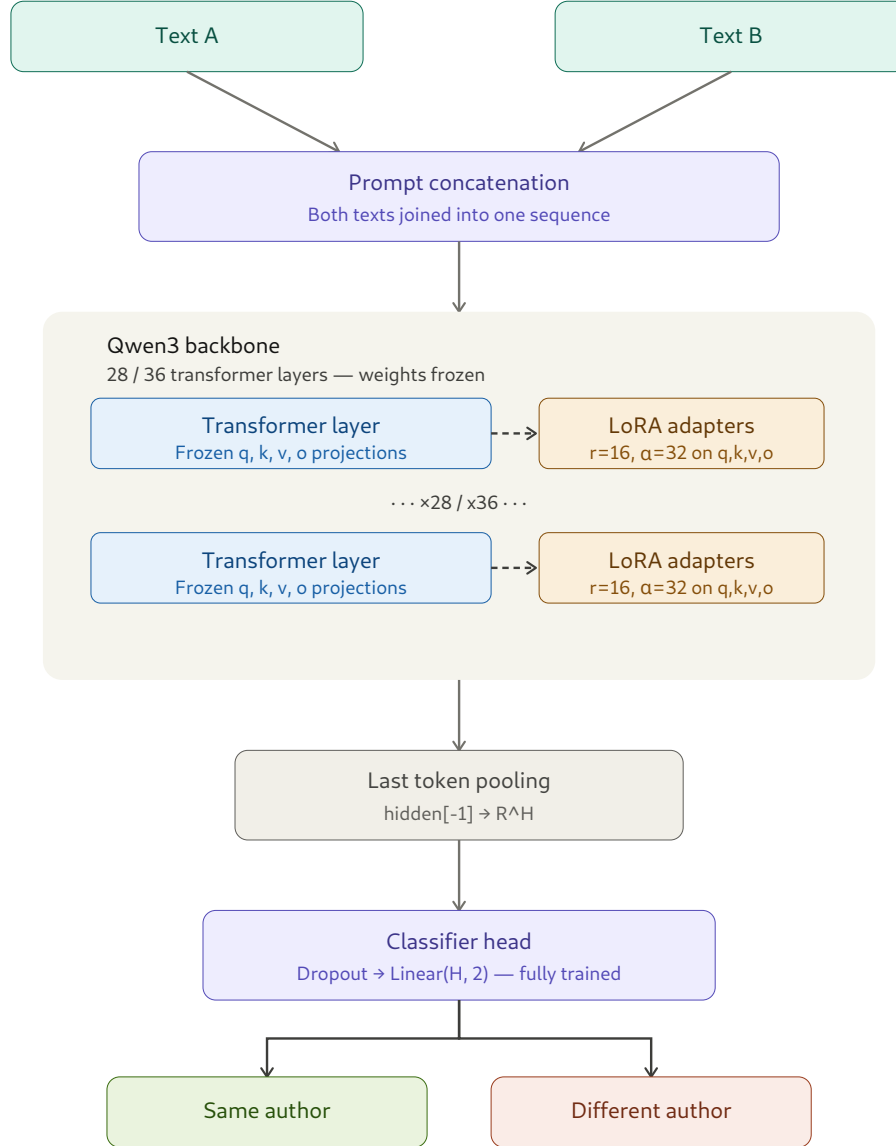


Figure 6.1: Architecture of the decoder classifier. Both texts are concatenated into a single prompted sequence and processed in one forward pass through the Qwen3 backbone. LoRA adapters are applied to all attention projection matrices. The hidden state of the final token is extracted, passed through dropout, and projected to a binary classification output.

As the optimization algorithm, the fused AdamW optimizer was employed. Compared to the standard AdamW implementation, fused AdamW offers improved efficiency on CUDA-enabled hardware by performing weight updates within a single fused kernel rather than through multiple separate element-wise operations. This reduces kernel launch overhead and improves training throughput.

The weight decay coefficient was set to 0.01, which represents a mild but standard regularization setting. This helps prevent the relatively small number of trainable parameters from overfitting to noise in the training data. The learning rate was set to 10^{-4} , corresponding to the upper range commonly used in LoRA fine-tuning. This higher learning rate is

justified by the fact that only the adapter matrices and classification head are updated during training, while the base model parameters remain frozen.

Since the classification head is initialized randomly, a learning rate scheduler was introduced to stabilize early training. The scheduler combines a linear warmup phase with linear decay, controlled by a warmup ratio of 0.05. Consequently, during the first 5% of total training steps, the learning rate increases gradually from zero to the target value of 10^{-4} , after which it decreases linearly for the remainder of training. This strategy improves optimization stability and prevents abrupt parameter updates in the early stages of fine-tuning.

All configurations were trained for 3 epochs. Although this number may appear relatively low in comparison to full-model fine-tuning, it is sufficient in the present setting because only the LoRA adapter parameters and the classification head are updated during training. The backbone model remains frozen throughout the process, which substantially reduces the number of trainable parameters and allows the model to converge more rapidly.

6.2.1 Training Runs

The training loss curves shown in Figure 6.2 cover more than 15,000 optimisation steps for all four evaluated configurations, corresponding to two backbone sizes and two loss functions. On the linear scale (left), all runs exhibit a steep decline during the first 1,000–2,000 steps, indicating rapid initial convergence, after which the curves gradually stabilise into a plateau.

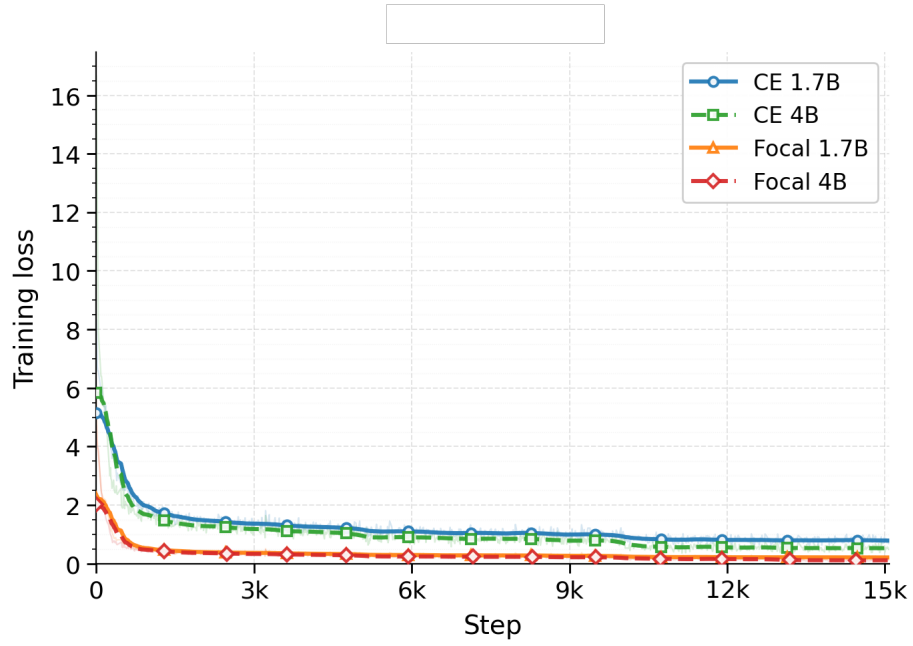
The logarithmic scale (right) reveals additional structure that is less visible in the linear representation. In particular, the cross-entropy configurations (CE 1.7B and CE 4B) converge to noticeably higher absolute loss values (approximately 0.7 and 0.5, respectively) than the focal loss variants (approximately 0.2 and 0.1). This difference should not be interpreted as evidence of superior generalisation by focal loss. As discussed in Section 3.1.7, focal loss applies a modulating factor $(1 - p_t)^\gamma$, which reduces the contribution of well-classified examples and therefore lowers the absolute scale of the loss. Consequently, the absolute magnitudes of cross-entropy and focal loss are not directly comparable.

Figure 6.3 presents the validation ROC-AUC over the full training duration of 15,000 steps. The left side of the figure shows the complete training trajectory, where all configurations demonstrate a sharp improvement during the first 1,000 steps, reflecting rapid early learning. The right side focuses on the plateau region beginning around step 2,000, where the differences between configurations become more distinguishable. This zoomed view highlights the relative stability of the models after convergence and enables a clearer comparison of their final validation performance.

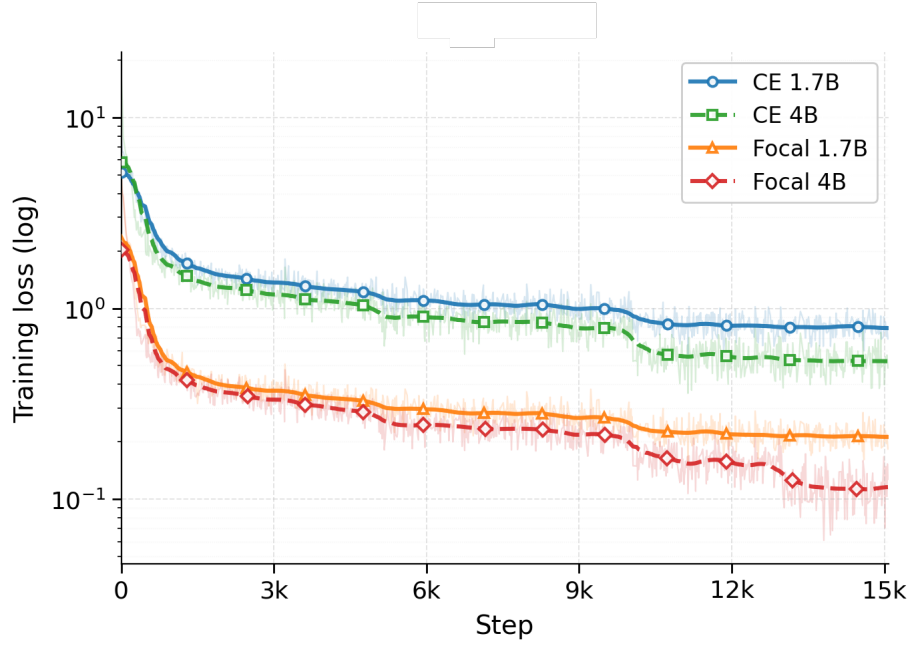
6.3 Evaluation

The results presented in Table 6.1 for the custom dataset and Table 6.2 for the PAN21 benchmark demonstrate that all four evaluated configurations achieve strong discriminative performance. On the custom dataset, ROC-AUC values range from 0.9882 to 0.9903, while on the PAN21 benchmark they range from 0.9348 to 0.9462, confirming that all variants perform reliably across both in-domain and cross-domain evaluation settings.

A clear scaling effect is visible across both benchmarks: increasing the backbone size from Qwen3-1.7B to Qwen3-4B consistently improves performance across all reported metrics, regardless of the loss function used. This observation contrasts with the findings of the

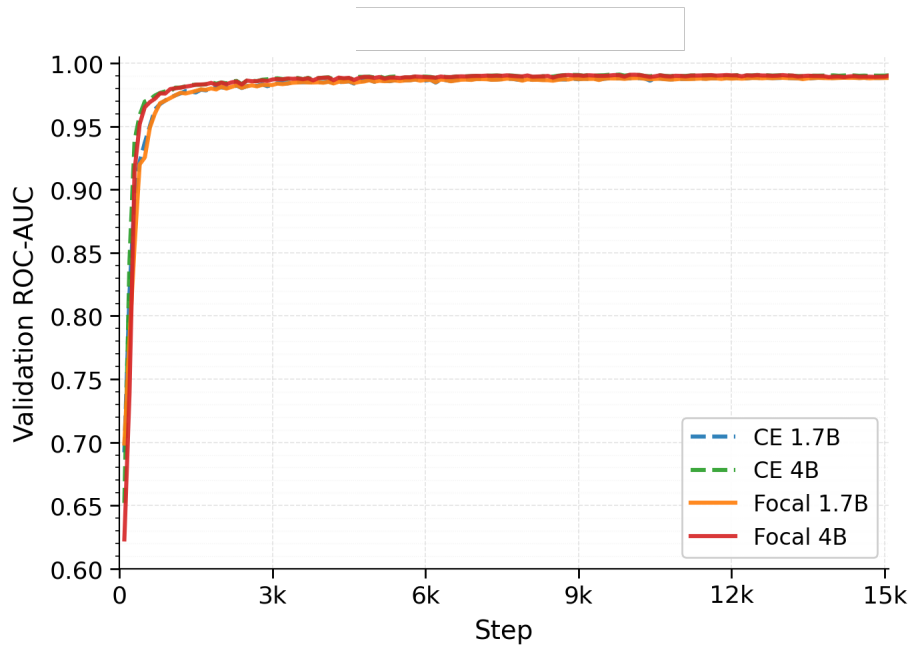


(a) Linear scale.

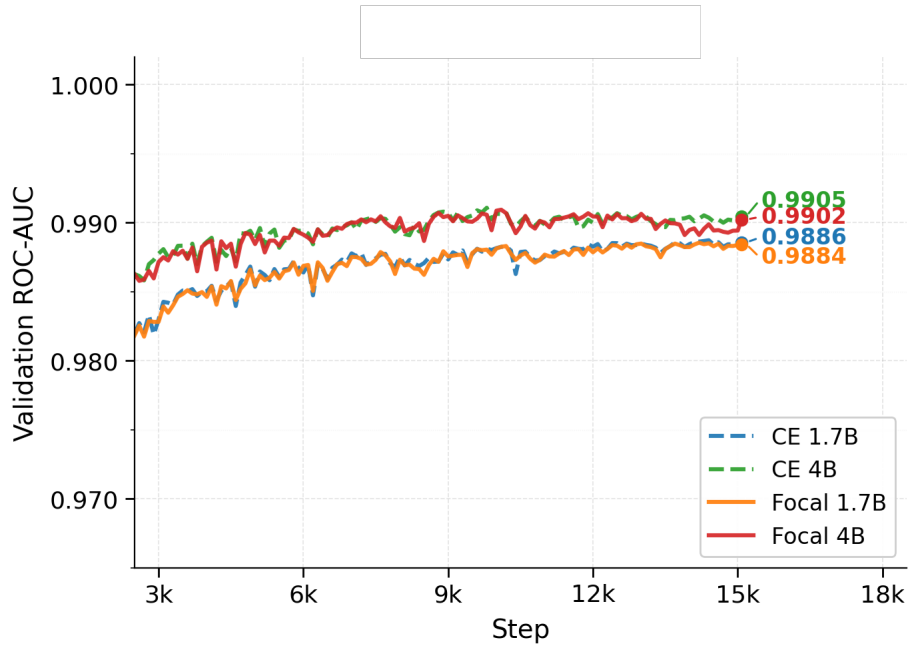


(b) Log scale.

Figure 6.2: Training loss curves for the decoder-based model across four configurations (cross-entropy vs. focal loss; 1.7B vs. 4B backbone).



(a) Full training run.



(b) Plateau (zoomed).

Figure 6.3: Validation ROC-AUC curves for the decoder-based model across four configurations: cross-entropy loss with 1.7B and 4B parameter models (CE 1.7B, CE 4B), and focal loss with the same model sizes (Focal 1.7B, Focal 4B).

preliminary prompting experiments in Section 4.3.2, where the larger 4B model performed significantly worse than the smaller 1.7B variant. The discrepancy suggests that fine-tuning enables the larger model to better exploit its higher representational capacity, whereas in zero-shot prompting it may be more sensitive to prompt formulation or less aligned with the task.

The contribution of focal loss is comparatively minor. When paired with the 4B backbone, focal loss improves ROC-AUC by only 0.0003 and MCC by 0.0003 over cross-entropy on the custom dataset, differences that remain within expected run-to-run variation. A similarly marginal effect is observed for the 1.7B backbone. These results indicate that, under the moderate class imbalance present in this dataset, focal loss provides no substantial advantage over standard cross-entropy once the model approaches saturation performance.

The best overall configuration on both benchmarks is Qwen3-4B trained with focal loss. On the custom dataset, this model achieves $AUC = 0.9903$, $EER = 4.79\%$, $F_1 = 0.9523$, and $MCC = 0.9042$. On the PAN21 benchmark, it reaches $AUC = 0.9462$, $EER = 12.84\%$, $F_1 = 0.8739$, and $MCC = 0.7410$, demonstrating the strongest generalisation among all evaluated configurations.

Table 6.1: Evaluation results on the custom test set (17,062 samples). Best result per column is in **bold**.

Model	AUC	EER	F1 ^a	MCC ^a	Bal. Acc ^a
Qwen3-1.7B (CE)	0.9882	0.0556	0.9450	0.8886	0.9440
Qwen3-1.7B (Focal)	0.9887	0.0556	0.9448	0.8889	0.9444
Qwen3-4B (CE)	0.9900	0.0481	0.9520	0.9039	0.9519
Qwen3-4B (Focal)	0.9903	0.0479	0.9523	0.9042	0.9521

^a Reported at the best-F1 threshold.

Table 6.2: Evaluation results on the PAN21 test set (19,999 samples). Best result per column is in **bold**.

Model	AUC	EER	F1 ^a	MCC ^a	Bal. Acc ^a
Qwen3-1.7B (CE)	0.9362	0.1411	0.8603	0.7101	0.8532
Qwen3-1.7B (Focal)	0.9348	0.1441	0.8587	0.7097	0.8541
Qwen3-4B (CE)	0.9455	0.1300	0.8719	0.7363	0.8671
Qwen3-4B (Focal)	0.9462	0.1284	0.8739	0.7410	0.8696

^a Reported at the best-F1 threshold.

The confusion matrices shown in Figure 6.4 provide further insight into the behaviour of the best-performing model at its optimal F_1 threshold. On the custom test set ($\tau^* = 0.4932$), the model achieves a false negative rate of 4.2% and a false positive rate of 5.4%, indicating a well-balanced decision boundary with only a slight bias toward precision. On the PAN21 benchmark ($\tau^* = 0.5567$), the false negative rate increases to 9.6%, while the false positive rate rises more sharply to 16.5%. This asymmetric degradation suggests that cross-domain generalisation affects negative-pair discrimination more severely than positive-pair recognition.

The shift in the optimal decision threshold between the two datasets further indicates a distribution mismatch between the custom dataset and PAN21. This threshold drift may

reflect differences in stylistic variability, domain composition, or semantic structure, and suggests that additional calibration or domain adaptation strategies may be beneficial for improving generalisation robustness.

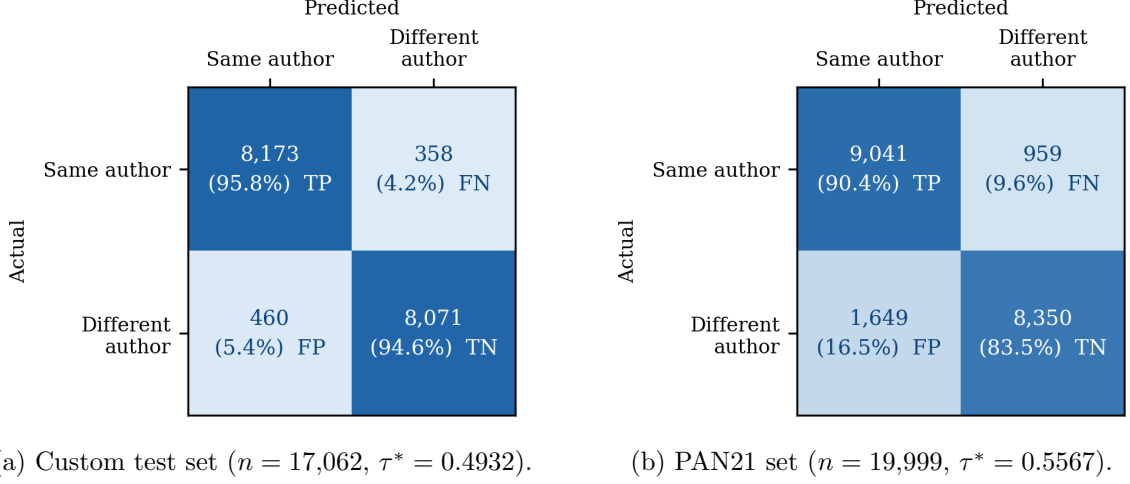


Figure 6.4: Confusion matrices of the best model (QWEN3-4B, Focal loss) at the best- F_1 threshold τ^* on the custom test set and the PAN21 held-out benchmark.

Comparison with State of the Art

Table 6.3 compares the proposed models against top-performing models from the PAN21 competition on their PAN21 test set (described in Section 4.1.6) on the authorship verification task. It is important to note that all prior systems were trained directly on the PAN21 calibration data, whereas both models proposed in this work were trained exclusively on the custom multi-source dataset described in Section 4.2. The evaluation on PAN21, therefore, constitutes a strictly out-of-domain generalisation test for our approaches.

Table 6.3: Prior systems (upper block) were trained on PAN21 calibration data (large or small track) and evaluated on the PAN21 open-set test set. Our models (lower block) used the custom dataset only, with PAN21 serving as a held-out out-of-domain benchmark.

System	Training set	AUC
boenninghoff21 [24]	PAN21 large	0.9869
weerasinghe21 [24]	PAN21 large	0.9719
embarcaderoruiz21 [24]	PAN21 large	0.9697
menta21 [24]	PAN21 large	0.9635
embarcaderoruiz21 [24]	PAN21 small	0.9470
tyo21 [55]	PAN21 large	0.8275
ROBERTA (ours)	Custom	0.9335
QWEN3-4B (ours)	Custom	0.9462

Despite this disadvantage, the decoder approach, specifically the Qwen3-4B trained with Focal loss, achieves an AUC of 0.9462, placing it competitively between the top small-set

submission and the large-set systems. The RoBERTa model based on encoder approach (described in Section 5), reaches an AUC of 0.9335, which, while lower, remains a strong result considering the domain shift and the absence of any PAN21 training signal. Both results suggest that the semantics-agnostic training strategy adopted in this work, which explicitly discourages reliance on topical similarity, promotes generalisation to unseen domains and author populations.

6.4 Experiment Conclusion

The experiments presented in this section demonstrate that decoder-only language models, when adapted using LoRA and formulated as prompted cross-encoders, provide an effective architecture for the authorship verification task. Fine-tuning the Qwen3-1.7B and Qwen3-4B models on the custom verification dataset resulted in strong in-domain performance, with the best configuration achieving $MCC = 0.9042$ on the custom test set (Table 6.1) and $MCC = 0.7410$ on the held-out PAN21 benchmark (Table 6.2).

The cross-encoder formulation, in which both texts are concatenated into a single structured prompt and processed jointly, proved to be both practical and effective. This validates the architectural hypothesis introduced in Section 6.1, namely that decoder-only transformers can successfully model cross-document dependencies by allowing tokens in one text to attend to the full context of the other. Such interaction is particularly beneficial in authorship verification, where relational stylistic cues between paired documents are critical.

Furthermore, the combination of this cross-encoder design with LoRA adaptation applied to the query, key, value, and output projection matrices enabled parameter-efficient fine-tuning while preserving strong task performance. These results confirm that decoder-based large language models can serve as highly competitive backbone architectures for authorship verification, even when adapted with lightweight fine-tuning strategies.

6.5 Application

To demonstrate the functionality and practical applicability of the approaches presented in this chapter, an application was developed that integrates both the decoder-based model described in Chapter 6 and the encoder-based model described in Chapter 5. The application is designed to be accessible both via a command-line interface and through an API, enabling flexible usage in different deployment scenarios.

6.5.1 Inference of the models

Inference is performed using the standard PyTorch backend. The inference methods are decorated with `@torch.no_grad()`, which disables gradient tracking. As a result, no back-propagation graph is built, so inference is faster and uses less memory. The model is moved to CUDA when a GPU is available; otherwise, it runs on CPU.

6.5.2 Application Structure

The application pipeline is structured into three main components: an input parser, a decoder-based backend, and an encoder-based backend. Each backend encapsulates a distinct inference pipeline corresponding to the respective model architecture.

The input parser is responsible for processing user input and preparing it in a format suitable for model inference. Based on the selected configuration, the request is then routed to either the decoder or encoder backend, where the appropriate model is executed to produce the final prediction.

The *DecoderBackend* is responsible for preparing the input as described in data preparation in Chapter 6. The backbone builds a single prompt and splits the token budget across both texts under the configuration file. Then the input is used for a forward pass through the model shown in Figure 6.1. At the end, the backend will compute the softmax of the logits and return the probability at index 1.

The second option for the backbone is the *EncoderBackend*, which will prepare and tokenise the texts separately, as described in Chapter 5. The model is then processed through the model shown in Figure 5.3. We take logits, compute softmax, and return the class-1 probability, as in DecoderBackend.

6.5.3 Runtime Performance Comparison

To better understand the practical advantages and limitations of the individual approaches, we additionally evaluated the inference speed of the trained models. This evaluation was performed using a separate benchmarking script on the machine described in Appendix C.1. Since the goal of this experiment was to compare the computational cost of the model architectures rather than the effect of a particular dataset, synthetic input data were used. The inputs followed the same configuration as during training, with the sequence length fixed to 512 tokens.

Table 6.4: Inference speed comparison between encoder and decoder approaches. Measured on device C.1, seq. len. 512, synthetic data.

Approach	Model	Loss	Params	Latency BS=1 (ms)	Throughput (p/s)	GPU Mem. BS=1 (MB)
Encoder	BERT-base	CE	257M	8.04	209.8	2010
		Focal	257M	8.03	209.8	2010
		SupCon	257M	8.04	209.4	2010
	RoBERTa-large	CE	829M	19.22	68.8	6381
		Focal	829M	19.16	69.1	6381
		SupCon	829M	19.23	68.7	6381
Decoder	Qwen3-1.7B	CE	1721M	16.48	105.2	3318
		Focal	1721M	16.49	105.2	3318
	Qwen3-4B	CE	4022M	28.02	45.1	7775
		Focal	4022M	27.97	45.1	7775

For each model, we measured the average inference latency for a batch size of one, the achieved throughput in processed pairs per second, and the peak GPU memory usage. The results are reported in Table 6.4. The comparison shows that the encoder-based models provide lower latency and lower memory consumption than the larger decoder-based models. However, Qwen3-1.7B achieves competitive latency despite having substantially more parameters than BERT-base, while Qwen3-4B is the slowest and most memory-demanding

model in the comparison. The choice of loss function has only a negligible effect on runtime performance, since it changes the training objective but not the inference architecture.

Chapter 7

Conclusion

This thesis investigated authorship verification based on stylometric analysis and modern machine learning methods. The theoretical part first introduced the fundamental concepts of biometrics, stylometry, and methodological frameworks used in stylometric analysis in Chapter 2. Subsequently, Chapter 3 presented the machine learning concepts and model architectures that formed the basis of the proposed approaches. The thesis also reviewed existing work combining stylometry and machine learning, with particular focus on transformer-based methods for authorship verification in Section 3.4. This analysis showed that most existing approaches rely primarily on BERT or RoBERTa encoders, often without substantial modifications to dataset construction or classifier architecture.

Based on the identified limitations in dataset preparation, this work proposed a custom pipeline for constructing a semantic-agnostic authorship verification dataset. The pipeline was designed according to two main principles: positive pairs should be semantically more distant, while negative pairs should be semantically closer, as illustrated in Figure 4.2. During dataset construction, several publicly available datasets described in Section 4.1 were used. By applying softmax roulette selection for hard negative pair mining and random sampling for positive pair mining, a total of 679,354 text pairs was created. The purpose of this dataset was to encourage models to rely more strongly on lexical and syntactic features, discussed in Section 2.4.1, while reducing their dependence on semantic similarity.

As a preliminary experiment, an LLM-based zero-shot pipeline was implemented to determine whether two texts were written by the same author. Several experiments were conducted using Qwen3, DeepSeek, and GPT-OSS models, with the results reported in Figure 4.2. The results indicate that the baseline discriminative ability of these models was generally higher on the PAN21 dataset, with the exception of the Qwen3:1.7B model. The best zero-shot performance was achieved by the DeepSeek model, which obtained approximately 0.21 MCC on the custom dataset and approximately 0.42 MCC on the PAN21 benchmark. The difference between these results suggests that the proposed custom dataset represents a more challenging evaluation setting for authorship verification.

The first proposed approach, introduced in Chapter 5, was based on an encoder architecture similar to the transformer-based methods discussed in Section 3.4. In contrast to existing approaches, two main extensions were introduced: co-attention injections and a style regularisation branch. Since both input streams share the same encoder weights, the co-attention mechanism enables information exchange between the two streams and supports the generation of more reliable representations for classification. The style reconstruction branch was designed to penalise representations that rely too strongly on semantic information and to encourage the extraction of stylistic features. Experimental results on

the custom dataset achieved ROC-AUC scores between 0.91 and 0.98. In the cross-domain evaluation on the PAN21 benchmark, the models achieved ROC-AUC scores between 0.85 and 0.93. The best encoder-based result was obtained by the RoBERTa-large model trained with cross-entropy loss, reaching approximately 0.98 ROC-AUC on the custom dataset and approximately 0.93 ROC-AUC on the PAN21 benchmark.

The second proposed approach, presented in Chapter 6, investigated decoder-based authorship verification and directly followed the preliminary experiment described in Section 4.3. In this approach, smaller decoder models, specifically Qwen3 1.7B and Qwen3 4B, were fine-tuned using LoRA adapters, as illustrated in Figure 6.1. The models were trained for three epochs using both cross-entropy loss and focal loss. Based on the results reported in Section 6.3, the decoder-based models achieved approximately 0.988–0.990 ROC-AUC on the custom dataset and approximately 0.936–0.946 ROC-AUC on the PAN21 benchmark. The best overall model was Qwen3-4B trained with focal loss, achieving 0.9903 ROC-AUC on the custom dataset and 0.9462 ROC-AUC on the PAN21 benchmark.

The comparison presented in Section 6.3 shows that the decoder-based approach achieved the strongest verification performance. In particular, the Qwen3-4B model trained with focal loss outperformed the other evaluated configurations on both the custom test dataset and the cross-domain PAN21 benchmark. However, the encoder-based approach remains a relevant and practical alternative in scenarios where inference speed, memory consumption, and computational efficiency are important, mainly due to its significantly smaller model size and lower hardware requirements.

In summary, this thesis contributes a complete authorship verification framework based on stylometry and modern transformer architectures. The main contributions include the construction of a semantic-agnostic dataset, the design and evaluation of an encoder-based architecture with co-attention and style regularisation, and the investigation of decoder-based models fine-tuned with LoRA. The experimental results demonstrate that decoder-based models, particularly Qwen3-4B trained with focal loss, achieved the strongest verification performance across both the custom dataset and the PAN21 benchmark. At the same time, the encoder-based approach remains a practical alternative for scenarios where computational efficiency and inference speed are important. These findings suggest that transformer decoders represent a promising but still underexplored direction for authorship verification.

Future work could extend this research by evaluating additional decoder-based models, exploring different LoRA configurations, and analysing the influence of dataset composition on cross-domain robustness. Another important direction is a more detailed comparison between encoder-based and decoder-based approaches under practical deployment constraints, such as inference time, memory usage, and hardware requirements. The results of this thesis were also submitted to the IJCB conference.

Bibliography

- [1] ABDI, H. and WILLIAMS, L. J. Principal component analysis. *WIREs Computational Statistics*, 2010, vol. 2, no. 4, p. 433–459. Available at: <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.101>.
- [2] AI COMPETENCE. *Stylometric AI: How It Detects Writing You Didn't Author*. N.d. Available at: <https://aicompetence.org/stylometric-ai-how-it-detects-writing-you-didnt-author/>. Accessed: 2025-12-24.
- [3] BAGNALL, D. Author identification using multi-headed recurrent neural networks. *ArXiv preprint arXiv:1506.04891*, 2015.
- [4] BOYKIS, V. What are embeddings. *10.5281/zenodo*, 2023, vol. 8015029.
- [5] CHICCO, D. and JURMAN, G. The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification. *BioData mining*. Springer, 2023, vol. 16, no. 1, p. 4.
- [6] CORBARA, S. and MOREO, A. Forging the forger: An attempt to improve authorship verification via data augmentation. *IEEE Access*. IEEE, 2024, vol. 12, p. 171911–171925.
- [7] DEEPSEEK AI. *DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models*. 2025.
- [8] DELAC, K. and GRGIC, M. A survey of biometric recognition methods. In: *Proceedings. Elmar-2004. 46th International Symposium on Electronics in Marine*. 2004, p. 184–193.
- [9] DEVLIN, J.; CHANG, M.-W.; LEE, K. and TOUTANOVA, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. Available at: <https://arxiv.org/abs/1810.04805>.
- [10] DRAHANSKÝ, M.; ORSÁG, F.; DVOŘÁK, R.; HÁJEK, J.; VÁŇA, J. et al. *Biometrie*. 1. vydáníth ed. Brno: Computer Press, s.r.o, 2011. 294 p. ISBN 978-80-254-8979-6.
- [11] FARIAS, S. T.; CHAND, V.; BONNICI, L.; BAYNES, K.; HARVEY, D. et al. Idea density measured in late life predicts subsequent cognitive trajectories: Implications for the measurement of cognitive reserve. *Journals of Gerontology Series B: Psychological Sciences and Social Sciences*. Oxford University Press, 2012, vol. 67, no. 6, p. 677–686.

- [12] GRAVES, A. Long Short-Term Memory. In: *Supervised Sequence Labelling with Recurrent Neural Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, p. 37–45. Available at: https://doi.org/10.1007/978-3-642-24797-2_4.
- [13] HALLIN, A. and WIKSTAD, T. *Stylometry in Academia: Estimating Authorial Contributions in Computer Science Bachelor Theses*. 2024. Bachelor’s Thesis. KTH Royal Institute of Technology. Available at: <https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-351179>. TRITA-EECS-EX 2024:384.
- [14] HOLLINGSWORTH, C. D. *Syntactic Stylometry: Using Sentence Structure for Authorship Attribution*. Athens, Georgia, 2012. Master’s Thesis. University of Georgia.
- [15] HU, E. J.; SHEN, Y.; WALLIS, P.; ALLEN ZHU, Z.; LI, Y. et al. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. Available at: <https://arxiv.org/abs/2106.09685>.
- [16] HUANG, B.; CHEN, C. and SHU, K. *Authorship Attribution in the Era of LLMs: Problems, Methodologies, and Challenges*. 2025. Available at: <https://arxiv.org/abs/2408.08946>.
- [17] HUANG, W. and GRIEVE, J. Authorial language models for AI authorship verification. *CEUR Workshop Proceedings*. RWTH Aachen University, august 2024, vol. 3740, p. 2645–2652. ISSN 1613-0073. Conference and Labs of the Evaluation Forum : Information Access Evaluation meets Multilinguality, Multimodality, and Visualization, CLEF 2024 ; Conference date: 09-09-2024 Through 12-09-2024.
- [18] INNOVATRICS. *Biometrics - Definition, FAQs* <https://www.innovatrics.com/glossary/biometrics/>. 2024. Accessed: 2026-01-06.
- [19] JAIN, A.; ROSS, A. and PRABHAKAR, S. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2004, vol. 14, no. 1, p. 4–20.
- [20] JAKKULA, V. Tutorial on support vector machine (svm). *School of EECS, Washington State University*, 2006, vol. 37, 2.5, p. 3.
- [21] JORDAN, M. I. and MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. *Science*, 2015, vol. 349, no. 6245, p. 255–260. Available at: <https://www.science.org/doi/abs/10.1126/science.aaa8415>.
- [22] KARSDORP, F.; KESTEMONT, M. and RIDDELL, A. Stylometry and the Voice of Hildegard. In: *Humanities Data Analysis: Case Studies with Python*. Princeton, NJ: Princeton University Press, 2021. Available at: <https://www.humanitiesdataanalysis.org/stylometry/notebook.html>.
- [23] KESTEMONT, M.; MANJAVACAS, E.; MARKOV, I.; BEVENDORFF, J.; WIEGMANN, M. et al. Overview of the Cross-Domain Authorship Verification Task at PAN 2020. In: *CLEF 2020 Working Notes*. CEUR-WS.org, 2020, vol. 2696. CEUR Workshop Proceedings. Available at: https://ceur-ws.org/Vol-2696/paper_264.pdf.

- [24] KESTEMONT, M.; MANJAVACAS, E.; MARKOV, I.; BEVENDORFF, J.; WIEGMANN, M. et al. Overview of the Cross-Domain Authorship Verification Task at PAN 2021. In: FAGGIOLI, G.; FERRO, N.; JOLY, A.; MAISTRO, M. and PIROI, F., ed. *Working Notes Papers of the CLEF 2021 Evaluation Labs*. September 2021, vol. 2936. CEUR Workshop Proceedings. ISSN 1613-0073. Available at: <https://ceur-ws.org/Vol-2936/paper-147.pdf>.
- [25] KHALID, U.; BEG, M. and ARSHAD, M. *RUBERT: A Bilingual Roman Urdu BERT Using Cross Lingual Transfer Learning*. February 2021.
- [26] KHAN, S.; NASEER, M.; HAYAT, M.; ZAMIR, S. W.; KHAN, F. S. et al. Transformers in vision: A survey. *ACM computing surveys (CSUR)*. ACM New York, NY, 2022, vol. 54, 10s, p. 1–41.
- [27] KHOSLA, P.; TETERWAK, P.; WANG, C.; SARNA, A.; TIAN, Y. et al. Supervised contrastive learning. *Advances in neural information processing systems*, 2020, vol. 33, p. 18661–18673.
- [28] KULKARNI, N.; MAHAJAN, S. and BANGAR, G. IRJET | Machine Learning Algorithms for Classification of Various Stages of Alzheimer’s Disease: A review. *IRJET*, august 2020, vol. 7, p. 817–824.
- [29] LAGUTINA, K.; LAGUTINA, N.; BOYCHUK, E.; VORONTSOVA, I.; SHLIAKHTINA, E. et al. A Survey on Stylometric Text Features. In: FRUCT. *Proceedings of the 25th Conference of Open Innovations Association (FRUCT)*. Helsinki, Finland: [b.n.], November 2019, p. 184–195. ISSN 2305-7254.
- [30] LARAMÉE, F. D. Introduction to Stylometry with Python. *Programming Historian*, 2018, vol. 7. Available at: <https://doi.org/10.46430/phen0078>.
- [31] LI, M.; FUNG, B.; YEH, w.-c. and FACHKHA, C. How to Tame Pre-Trained Transformers for Authorship Verification. *Available at SSRN 5229510*, january 2025.
- [32] LIN, T.-Y.; GOYAL, P.; GIRSHICK, R.; HE, K. and DOLLÁR, P. *Focal Loss for Dense Object Detection*. 2018. Available at: <https://arxiv.org/abs/1708.02002>.
- [33] LIPPE, P. *Tutorial 6: Transformers and Multi-Head Attention* https://uvadlc-notebooks.readthedocs.io/en/latest/tutorial_notebooks/tutorial6/Transformers_and_MHAttention.html. 2024. UvA Deep Learning Tutorials.
- [34] LIU, Z. *Reuter_50_50* UCI Machine Learning Repository. 2006. DOI: <https://doi.org/10.24432/C5DS42>.
- [35] MANOLACHE, A.; BRAD, F.; BURCEANU, E.; BARBALAU, A.; IONESCU, R. et al. Transferring bert-like transformers’ knowledge for authorship verification. *ArXiv preprint arXiv:2112.05125*, 2021.
- [36] MILLETT, L. I. and PATO, J. N. Biometric recognition: Challenges and opportunities. National Academies Press, 2010.
- [37] MISRA, R. News Category Dataset. *ArXiv preprint arXiv:2209.11429*, 2022.

- [38] MONACO, J. V.; STEWART, J. C.; CHA, S.-H. and TAPPERT, C. C. Behavioral biometric verification of student identity in online course assessment and authentication of authors in literary works. In: *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. 2013, p. 1–8.
- [39] OPARA, C. *StyloAI: Distinguishing AI-Generated Content with Stylometric Analysis*. 2024. Available at: <https://arxiv.org/abs/2405.10129>.
- [40] OPENAI. *Gpt-oss-120b & gpt-oss-20b Model Card*. 2025. Available at: <https://arxiv.org/abs/2508.10925>.
- [41] PENG, J.; CHOO, K.-K. R. and ASHMAN, H. User profiling in intrusion detection: A review. *Journal of Network and Computer Applications*, 2016, vol. 72, p. 14–27. ISSN 1084-8045. Available at: <https://www.sciencedirect.com/science/article/pii/S1084804516301412>.
- [42] PROF.A.D. DONGARE, P. K. A. K. Introduction to Artificial Neural Network. *International Journal of Engineering and Innovative Technology (IJEIT)*, 2012. Available at: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=04d0b6952a4f0c7203577afc9476c2fcab2cba06>.
- [43] RAMNIAL, H.; PANCHOO, S. and PUDARUTH, S. Authorship Attribution Using Stylometry and Machine Learning Techniques. In: . January 2016, vol. 384, p. 113–125. ISBN 978-3-319-23035-1.
- [44] RASAMOELINA, A. D.; ADJAILIA, F. and SINČÁK, P. A Review of Activation Function for Artificial Neural Network. In: *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. 2020, p. 281–286.
- [45] REIMERS, N. and GUREVYCH, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2019.
- [46] SCHLER, J.; KOPPEL, M.; ARGAMON, S. and PENNEBAKER, J. W. Effects of age and gender on blogging. In: *AAAI spring symposium: Computational approaches to analyzing weblogs*. 2006, vol. 6, p. 199–205.
- [47] SCHMIDT, R. M. *Recurrent Neural Networks (RNNs): A gentle Introduction and Overview*. 2019. Available at: <https://arxiv.org/abs/1912.05911>.
- [48] SEROUSSI, Y.; ZUKERMAN, I. and BOHNERT, F. Authorship attribution with topic models. *Computational Linguistics*. MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , 2014, vol. 40, no. 2, p. 269–310.
- [49] SOLÓRZANO, J.; MIJANGOS, V.; PIMENTEL, A.; LÓPEZ ESCOBEDO, F.; MONTES, A. et al. Authorship Verification by combining SVMs with kernels optimized for different feature categories. *Working Notes Papers of the CLEF*, 2015.
- [50] STAMATATOS, E. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 2009, vol. 60, no. 3, p. 538–556. Available at: <https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.21001>.

- [51] STOLERMAN, A.; FRIDMAN, L.; GREENSTADT, R.; BRENNAN, P. and JUOLA, P. Active Linguistic Authentication Using Real-Time Stylometric Evaluation for Multi-Modal Decision Fusion. In: January 2014, vol. 433, p. 165–183. ISBN 978-3-319-12567-1.
- [52] TEAM, Q. *Qwen3 Technical Report*. 2025. Available at: <https://arxiv.org/abs/2505.09388>.
- [53] TENNEY, I.; DAS, D. and PAVLICK, E. BERT Rediscovered the Classical NLP Pipeline. In: KORHONEN, A.; TRAUM, D. and MÀRQUEZ, L., ed. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, p. 4593–4601. Available at: <https://aclanthology.org/P19-1452/>.
- [54] TRIPATHI, A. *What is the main difference between RNN and LSTM?* Data Science Duniya. 2. July 2021. Available at: <https://ashutoshtriplathi.com/2021/07/02/what-is-the-main-difference-between-rnn-and-lstm-nlp-rnn-vs-lstm/>. Accessed: 2026-01-04.
- [55] TYO, J.; DHINGRA, B. and LIPTON, Z. C. Siamese Bert for Authorship Verification. In: *CLEF (working notes)*. 2021, p. 2169–2177.
- [56] VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L. et al. Attention Is All You Need. *CoRR*, 2017, abs/1706.03762. Available at: <http://arxiv.org/abs/1706.03762>.
- [57] WANG, Z.; CHU, Z.; DOAN, T. V.; NI, S.; YANG, M. et al. History, development, and principles of large language models: an introductory survey. *AI and Ethics*. Springer, 2025, vol. 5, no. 3, p. 1955–1971.
- [58] WU, Y.-c. and FENG, J.-w. Development and application of artificial neural network. *Wireless Personal Communications*. Springer, 2018, vol. 102, no. 2, p. 1645–1656.
- [59] YULE, C. U. *The statistical study of literary vocabulary*. Cambridge University Press, 2014.

Appendix A

Encoder model architecture

Figure A.1 illustrates the full Encoder-Approach architecture built on a BERT backbone. Dark orange blocks represent the co-attention injection modules, while light orange blocks correspond to the BERT encoder streams. The blue box denotes the style reconstruction branch, which is applied only during training. The final yellow block constitutes the network’s last part and serves as the classifier.

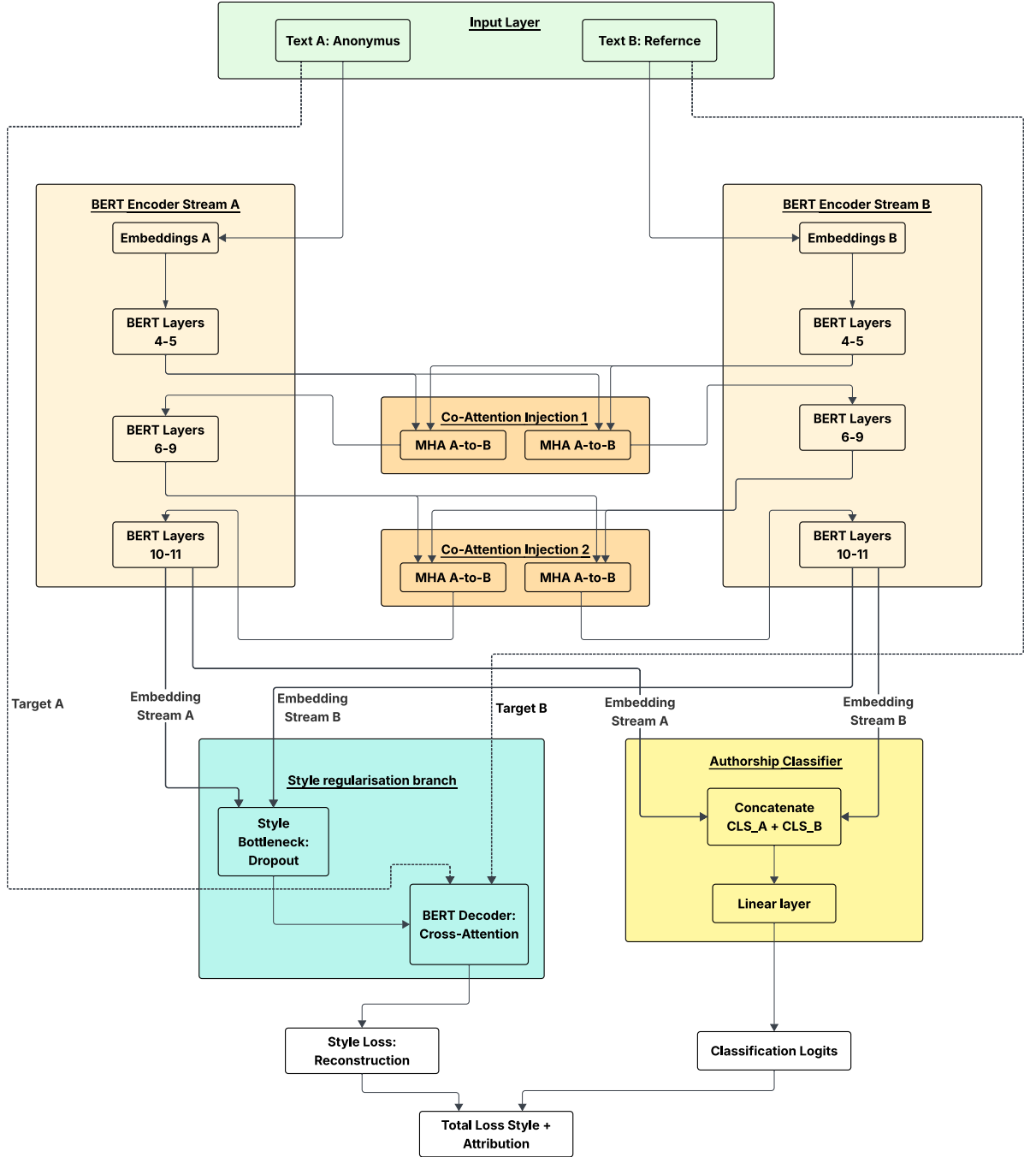


Figure A.1: Full model architecture of the Encoder Decoder model for BERT backbone.

Appendix B

Dataset overview: Pair example

B.1 Positive example

Text 1 :

Anglo-Dutch food-to-detergent group Unilever Plc, reporting third quarter results next Friday, is likely to have suffered from restructuring costs for its North American industrial cleaning business Diversey and from weak ice cream sales in a drab European summer, analysts said. Diversey, acquired in January from Molson for 360 million stg, was merged with Lever Industrial in October to form a combined business with annual turnover of 1.15 billion stg and a workforce of 13,000 in 60 different countries. Analysts predict third quarter pretax profits in a range of 760 to 820 million, against 786 million stg last time. The wide range reflects uncertainty about the size of redundancy payments and closures needed to reshape Diversey. Charges for Diversey could range from 15-30 million stg, with a further 20 million needed for restructuring other Unilever businesses. But the half-year dividend, paid in the third quarter, is forecast to rise sharply as a result of changes to dividend policy announced at the full year. Analysts expect a 10.3p dividend payout under a new policy whereby 35 percent of the previous year's total dividend would be paid in the stronger of either Dutch guilder or sterling so far this year. A 10.3p payout would represent a 46 percent increase on the previous half-year's 7.05p. For the nine month period analysts are forecasting pretax profits of about 1.9 billion pounds.

"European ice cream sales are not going to have the special benefit they had in 1995 or 1994," said John Campbell, analyst at Paribas. "European economies are generally weak, and German consumption if anything lower because of the tough economic climate there," he added. Michael Bourke at Panmure Gordon saw three factors behind the downturn. "The exceptional charge will be a bit higher, perhaps about 40 million compared with 20 million the year before, and ice cream sales in the third quarter will have been influenced a lot by the summer weather." "Last but not least is the currency factor, with the majority of Unilever profits in US dollars and the recent weakness in the dollar clearly having an impact." The consensus forecast EPS for 1996, taken from 24 brokers polled by Reuters, is 84.40p, rising to 92.30p in 1997. Six brokers downgraded their 1996 EPS forecast in October. Unilever's return on equity is currently running at 27.3 percent, against 66.57 percent for Unigate, the highest in the FTSE all-share food manufacturing sector. The company's shares reached a high of 1412 pence for the year in mid-September before closing down at 1283 on Friday. -- London Newsroom +44 171 542 6437

Text 2 :

British cider group Matthew Clark said on Tuesday fashionable "alcopop" drinks had helped to wipe a third off the sales of its premium brands -- Diamond White, K and Babydam -- in the first five months of the financial year. Matthew Clark has been hit the hardest by a surge in popularity for so-called alcopops -- alcoholic colas and lemonades -- which have proved a success with younger drinkers. Fierce price wars among rival drinks companies have also had an affect. Since Matthew Clark stunned the stock market in September with a profit warning that wiped more than 290 million pounds (\$466 million) off its market value, investors have badgered the group to explain how it intends to win back sales. In the long-awaited statement, Matthew Clark conceded that price wars had continued and that sales of draught cider in June and July, were 10 percent lower than during last year's hot summer. Premium brands saw sales slump 30 percent. The group's draught cider sales were 15 percent down across the whole summer -- although current trading in cider had since stabilised. "We expect the underlying growth within the cider market as a whole to resume," the company said. But it said it expected to hold its interim dividend at nine pence when it reported half year results ending October 31. The company's shares closed a sharp 17.5 pence higher at 317.5 pence, although far short of their year high of 804 pence. "The thing that's made them bounce is they've said they expect to maintain the dividend. But you've got to be cautious because there's no guide as to underlying profit levels," said one dealer.

The company has launched a review of its marketing strategy in branded drinks, mainly Diamond White and mainstream ciders Dry Blacthorn and Gaymers Olde English. The key findings will be released in January. In the meantime, the company said it would create the new position of marketing director, a main board appointment, and take on 40 staff in sales and marketing in its branded drinks division. Alcopops have proved especially popular with young drinkers, forging major inroads into the British drinks market. In May, research body Euromonitor said it would be worth about 170 million pounds this year after 100 million in 1995. Leading brands include alcoholic lemonade Hooper's Hooch, owned by Bass, and Two Dogs, marketed by Merrydown. Drinks industry leaders agreed in January to a voluntary code of practice aimed at limiting the promotion of alcopops after the drinks sparked a storm of controversy over whether they encouraged under-age drinking. In its last budget, the government imposed a heavy duty on ciders with an alcohol content over 7.5 percent from next year but left alcoholic colas and lemonades unaffected.

B.2 Negative example

Text 1 :

A festive calm settled over Hong Kong on Thursday as people prepared to welcome the Year of the Ox, the territory's last Lunar New Year celebration under British rule.

With the clock ticking towards transition to Chinese sovereignty at midnight on June 30, Hong Kong people took time out from hectic schedules for the Chinese equivalent of a Western Christmas or American Thanksgiving.

Last-minute shoppers dashed through malls, record numbers of travellers flowed across the border to visit relatives in China and children threw empty red "lai see" money pockets at each other on Hong Kong harbour's Star Ferry. "Kung Hei Fat Choi, may you be rich and make more money," said Ng Chak Kwong. "

Everybody wants this to be a good year. Too many people worry about everything ." But the cheerful greeting cards, office towers ablaze with lights, strings of red firecrackers and little orange trees offered few clues to Hong Kong's real state of mind on New Year's eve. This year's festivities were set against a backdrop of rising political tension after China's recent pledge to roll back some civil liberties laws following the handover, plans that sparked an outcry both at home and abroad. The Democratic Party, the territory's largest party, collected 20,000 signatures opposing the plans while the international community lodged official protests with China. Chief Executive-designate Tung Chee-hwa, the man who will take over from colonial Governor Chris Patten after the handover, last week defended the proposed legal changes as misunderstood, but few were convinced. Some of the families gathering for traditional New Year banquets on Thursday night said they would watch a televised annual address by Patten, even though the British influence is already considered virtually irrelevant. The last New Year's address by a British colonial governor could provide Patten with an ideal opportunity to lambast old foes in Beijing.

"What he says has no effect any more," said shopper Alice So. "What he says, what he did, what he's going to do, the Chinese government will not accept. It doesn't matter whether he's right or wrong." Most Hong Kong people seemed determined to ignore politics during the holiday. Four days of parades and fairs were planned, and spectacular fireworks would light up the sky on Saturday. According to Chinese astrologers, the coming year offered Hong Kong plenty of promise but obstacles as well. Unlike Western astrology, where months of the year are grouped into separate star signs, Chinese astrology hinges on the year of birth. That could mean bad news for Tung, who was born in 1937, the Year of the Ox. According to the astrologer at the South China Morning Post newspaper, Tung faced a difficult year marked by envy and accident and is advised to seek compromise. "He is likely to be caught between Hong Kong people and the Chinese government and may be the object of envy by mainland officials," warned astrologer Sung Siu-kwong. Patten, born under the sign of the Monkey in 1944, was warned about the emergence of a rival in mid-year and cautioned to draw a line between private and public involvements. "He will need to overcome obstacles that can stand in the way of the glorious withdrawal," Sung said.

Text 2 :

China has revived a call by Communist Party chief Jiang Zemin to wipe out poverty before the end of the century in a bid to placate its backward hinterland, analysts said. The text of a speech Jiang made in September calling for more funds to fight poverty was splashed across front pages of major newspapers on Monday. The coverage was aimed partly at bolstering the Communist Party's standing in impoverished but politically sensitive regions, one Beijing-based Western diplomat said. "It shows that the party leadership has not forgotten the poor," said the diplomat, who asked not to be identified. The reports were also timed to show concern ahead of the Lunar New Year, China's most important holiday, analysts said. Chinese leaders traditionally make highly publicised trips to poor areas to hand out gifts before the holiday which this year begins on February 7. "The poor can have a happy Lunar New Year, but after the holidays, life will still be difficult," one Chinese analyst said. "The gap between the rich and poor is becoming wider and there's more bitterness." China's eastern coastal provinces have boomed under nearly two decades of economic reform that have allowed some areas to get rich faster than others. Provinces in central and western China have lagged badly. Beijing

fears that the widening gap between rich and poor could spark social unrest and undermine its grip on power. Analysts said that was a major concern ahead of a crucial Communist Party congress scheduled for later this year and expected to set the course of national policy over the medium term. "The scale of poverty relief will be bigger this year," the Chinese analyst said. More people were in need of aid after being laid off from loss-making or insolvent state enterprises, he said. "They're playing up a 1996 speech to show the government is earnestly concerned about the problem and trying to do something about it," the analyst said. In the speech, Jiang urged the nation to "gnaw the hard bone" of poverty by pouring more funds into eliminating the problem by the end of the century. China says it reduced the number of people living in poverty by five million in 1995 but about 65 million people are still struggling to get by with incomes under the official poverty line of 530 yuan (\$64) a year. The central government currently spends 10.8 billion yuan a year on poverty relief and Beijing has pledged a further 1.5 billion yuan to build roads and water facilities in impoverished areas. During an inspection tour of some of China's poorest regions last September, Jiang tried to resurrect communist revolutionary fervour to fight poverty. He exhorted villagers and officials to resurrect the "glorious tradition" of the revolution that won the civil war for the communists in 1949. (\$1 = 8.3 yuan)

Appendix C

Machines used for training

C.1 Machine 1

- **CPU** : Intel(R) Core(TM) Ultra 9 285K
- **GPU** : NVIDIA GeForce RTX 5090 (32GB)
- **RAM** : 32GB

C.2 Machine 2

- **CPU** : Intel(R) Xeon(R) Silver 4314 CPU @ 2.40GHz
- **GPU1** : NVIDIA RTX A5000 (24GB)
- **GPU2** : NVIDIA RTX A5000 (24GB)
- **GPU3** : NVIDIA RTX A5000 (24GB)
- **GPU4** : NVIDIA RTX A5000 (24GB)
- **RAM** : 256GB

C.3 Machine 3

- **CPU** : AMD EPYC 9454 48-Core Processor
- **GPU** : NVIDIA H100 NVL 96GB
- **RAM** : 1.3TB