

Representation Learning and Deep Neural Approaches for International Trade Product Classification

Angga Wahyu Anggoro

A thesis submitted for the degree of Doctor of Philosophy

Cardiff University
September 2025

Abstract

International trade underpins the global economy, with trade volumes reflecting cross-border exchange, and tariffs function both as government revenue and as instruments of domestic market protection. The extensive documentation in international trade administration underlines the necessity of digitalisation to ensure the effectiveness of trade facilitation. The complexity of text-based trade administration presents challenges in optimising information processing, validation and utilisation. Employing language models has emerged as a practical paradigm for dealing with the complexities of natural language, presenting research opportunities for optimising automated machine learning tasks for trade document processing. This thesis first addressed the challenge of classifying thousands of product categories by applying effective preprocessing strategies to trade data containing heterogeneous product attributes. Contextualised representations were generated from selected modern pretrained language models, fine-tuned on domain-specific trade datasets, and subsequently fed into a novel hierarchical classification network. This pipeline approach shows substantial performance improvements. Second, supervised contrastive learning was implemented using a Siamese network to enhance sentence embeddings to represent trade product declarations. This approach facilitates flexibility for effective integration with machine learning classifiers and outperforms well-established methods for product classification. Third, to detect product misclassification, this thesis proposes a text anomaly detection framework based on autoencoders. Input representation from pretrained language models combined with autoencoders effectively identified contextualised anomalies under both semi-supervised and unsupervised learning settings. Finally, a bilingual product classification model was developed that incorporated cross-lingual keywords during model training. The proposed method aligns product representation and achieves superior performance in challenging low-resource language settings, utilising bilingual trade data. This thesis concludes that the integration of product representation from modern language models and deep learning architectures presents strong potential to automate trade processes. Additionally, this thesis could present as the groundwork for both methodological and practical aspects of machine learning-based systems in the trade domain.

Contents

Abstract.....	i
Contents.....	ii
List of Tables	vi
List of Figures	x
List of Abbreviations.....	xiv
Acknowledgements.....	xvi
Chapter 1 Introduction	1
1.1 International Trade	1
1.2 Challenges of Automating Tasks in International Trade	3
1.2.1 Trade Product Classification.....	3
1.2.2 Product Misclassification	4
1.2.3 Bilingual Languages Classification	5
1.3 Research Landscape in International Trade	5
1.3.1 Text Classification Task	5
1.3.2 Sentence Representation Learning	7
1.3.3 Text Anomaly Detection	7
1.3.4 Bilingual Classification Task	8
1.4 Research Aims, Questions, and Objectives	9
1.5 Research Contribution.....	11
1.6 Thesis Structure	14
Chapter 2 Literature Review: Foundations of Automated International Trade	
Classification Systems	16
2.1 International Trade Data.....	17
2.1.1 Harmonized Systems	17
2.1.2 Trade Declaration	19

2.1.3	Trade Datasets	20
2.2	Classification Task in Trade Transactions	22
2.2.1	Classification Methods	23
2.2.2	Classification Approaches	27
2.2.3	Text Preprocessing Techniques	30
2.2.4	Text Representation Methods	33
2.2.5	Trade Product Classification with Word Embeddings	43
2.3	Sentence Embedding in Trade Transactions	45
2.3.1	Trade Product Classification with Sentence Embeddings	47
2.4	Anomaly Detection in Trade Transactions	49
2.4.1	Anomaly Detection using Heterogeneous Data	51
2.4.2	Anomaly Detection using Text Data	53
2.4.3	Anomaly Detection with Autoencoder for Product Misclassification	55
2.5	Bilingual Classification in Trade Transactions	57
2.5.1	Monolingual Classification	57
2.5.2	Bilingual Classification	58
2.5.3	Cross-lingual Embeddings	59
2.5.4	Bilingual Classification with Keywords Integration	62
2.6	Summary	64

Chapter 3 Optimising the Text Classification Pipeline for Trade Product

Classification	65
3.1 Introduction	65
3.2 Methodology	70
3.2.1 Problem Formulation	71
3.2.2 Dataset	71
3.2.3 Data Preprocessing	72
3.2.4 Pretrained Language Models	74
3.2.5 Classification Approaches	79
3.2.6 Experimental Setting	83
3.3 Result and Discussion	86
3.3.1 Preprocessing Results	86
3.3.2 Transformers-based Models	92
3.3.3 Hierarchical Classification	96

3.3.4 Flat and Hierarchical Classification	97
3.4 Summary	98

Chapter 4 Enhancement of Trade Product Classification through Sentence Embeddings and Supervised Contrastive Learning 100

4.1 Introduction.....	100
4.2 Methodology.....	104
4.2.1 Text Encoder	105
4.2.2 Pooling Layers.....	105
4.2.3 Contrastive Learning	106
4.2.4 Samples Configuration	106
4.2.5 Multiple Negative Ranking Loss	108
4.3 Experiments Settings	112
4.3.1 Datasets	112
4.3.2 Model Configuration	113
4.3.3 Model Baselines	115
4.3.4 Experiments Evaluation.....	115
4.4 Results and Discussion	118
4.4.1 Embeddings Characteristics.....	119
4.4.2 Classification Performance.....	124
4.4.3 Components Evaluation	129
4.5 Summary.....	135

Chapter 5 Text Anomaly Detection for Trade Product Misclassification via Autoencoders 136

5.1 Introduction.....	136
5.2 Methodology.....	141
5.2.1 Semi-supervised and Unsupervised Learning with Autoencoder.....	141
5.2.2 Anomaly Score and Threshold	151
5.3 Experiments Settings	152
5.3.1 Datasets	152
5.3.2 Baseline Models	156
5.3.3 Evaluation Metrics	157
5.3.4 Experiments Settings	157

5.4	Result and Discussion	158
5.4.1	Semi-Supervised Approach	158
5.4.2	Unsupervised Approach	172
5.5	Summary	175
Chapter 6 Bilingual Product Classification through Keyword Integration in Low-Resource Trade Data		177
6.1	Introduction.....	177
6.2	Methodology.....	182
6.2.1	Problem Formulation	184
6.2.2	Keyword Extraction and Modelling	185
6.2.3	Dual Encoder.....	187
6.2.4	Fusion Layer.....	188
6.2.5	Deep Classification Layer.....	189
6.3	Experiments	189
6.3.1	Datasets	189
6.3.2	Experiments Settings	191
6.4	Results and Discussion	194
6.4.1	Model Performance	194
6.4.2	Embeddings Alignment.....	195
6.4.3	Ablation Study	198
6.4.4	Model Performance in Low-Resource Setting	199
6.5	Summary	201
Chapter 7 Conclusion.....		203
7.1	Revisiting Automation Tasks in International Trade	203
7.2	Contributions	206
7.3	Future Research.....	208
Bibliography		211

List of Tables

Table 2.1 Hierarchical structure of HS codes based on digit level.....	18
Table 2.2 Summary of datasets utilised in previous studies on international trade...	21
Table 2.3 Examples of differences between technical HS code technical definitions and commercial terminology.....	22
Table 2.4 Preprocessing techniques implemented in trade transactions.....	33
Table 2.5 A comparison of language models based on Transformer architecture. ...	42
Table 2.6 Key characteristics of sparse versus (non-)contextualised text representations.....	43
Table 3.1 The trade dataset statistics.....	72
Table 3.2 Preprocessing techniques used for trade data.....	73
Table 3.3 List of preprocessing strategies and techniques applied to trade transaction data.	74
Table 3.4 Dataset statistics across applied preprocessing strategies.....	74
Table 3.5 Selected transformer-based models with their corresponding tokenisation algorithms and vocabulary sizes.	76
Table 3.6 Comparison of tokenisation outputs from different Transformer pretrained models. The table shows how a sample product description is tokenized by various models, revealing differences in the form of tokens and unique numbers of tokens produced due to their distinct tokenization algorithms.	77
Table 3.7 Default hyperparameter configuration used for the trade product classification task.	84
Table 3.8 Baseline language models used to generate text representations.....	84
Table 3.9 Model accuracy results in the classification performance of transformer- based and baseline models across three preprocessing strategies.	86
Table 3.10 Number of tokens generated by pretrained language models for the training dataset (in millions) across different preprocessing strategies and their impact on model accuracy.	89
Table 3.11 Performance of fine-tuned Transformer-based language models across various metrics using preprocessing strategy PP-1 under flat classification.	93
Table 3.12 Number of classes in which each model (vertical columns) performs better or equal compared to other models (horizontal columns).	93

Table 3.13 Duration of training time (hours: minutes: seconds) and parameter sizes of pretrained language models. The results reflect how architectural differences and model size influence fine-tuning efficiency when processing datasets with different preprocessing strategies (trained on an NVIDIA Tesla P100 GPU).	95
Table 3.14 Model performance of fine-tuned DistilBERT using the proposed hierarchical classification method. The results highlight model performance at different HS code levels, with an increasing number of classes from Chapter (HS-2) to Sub-heading (HS-6).	97
Table 3.15 Model performance using the flat and hierarchical approach for HS-6 codes.	98
Table 4.1 The statistics of India and U.S. trade datasets available for the experiments.	113
Table 4.2 Comparison of similarity scores between intra-class and inter-class generated from each method.	124
Table 4.3 Comparison of the model classification performance on Dataset 1, demonstrating enhanced sentence embeddings combined with SVM and RF classifiers outperforming baseline models across key evaluation metrics.	125
Table 4.4 Statistical comparison between the best-performing configuration of the proposed method and baseline models on Dataset 1, presenting that the proposed method significantly outperformed the baseline methods, with statistically significant differences observed.	126
Table 4.5 Comparison of the model classification performance on Dataset 2, demonstrating enhanced sentence embeddings combined with SVM and RF classifiers outperforming baseline models across key evaluation metrics.	126
Table 4.6 Statistical comparison between the best-performing configuration of the proposed method and baseline models on Dataset 2, presenting that the proposed method significantly outperformed the baseline methods, with statistically significant differences observed.	127
Table 4.7 Classes with the substantial F1-score differences between the fine-tuned DistilBERT model and the proposed method.	129
Table 4.8 Comparison of the model classification performance under different numbers of positive samples per anchor.	130
Table 4.9 Comparison of the model classification performance when using three pooling strategies, including CLS token, mean pooling, and max pooling.	131

Table 4.10 Comparison of the model classification performance across varied loss function configurations.	132
Table 4.11 Comparison of the model classification performance under different hyperparameter configurations reporting evaluation metrics across varying learning rates, batch sizes, and epochs.	134
Table 5.1 OAE network layer specifications contain the encoder that includes two fully connected layers that compress the 768-dimensional input into a 384-dimensional latent representation using ReLU activations. The decoder symmetrically reconstructs the input and uses Tanh in the output layer.	146
Table 5.2 Examples of the transactions in the datasets. Products with a 0% (duty-free) rate are labelled as normal. In contrast, products with higher duty rates are treated as anomalous in our experiment, as they will be deliberately assigned the same HS code used for normal data.	154
Table 5.3 Examples of noise steps treatment to a product description.	155
Table 5.4 Distribution of normal and anomalous samples in the dataset. Anomalous samples in the testing set are a variant proportion as a percentage of the normal data in the testing split.	156
Table 5.5 Experimental hyperparameter configuration.	158
Table 5.6 presents the AUC performance of baseline models versus the proposed autoencoder-based models, the models were tested under various anomaly rates and contamination parameters using semi-supervised learning with 10-fold CV.	160
Table 5.7 Statistical comparison of the OAE against baseline models and other autoencoder variants in semi-supervised learning settings.	161
Table 5.8 Presents the AUC performance of shallow and deep autoencoder layers models, the models were tested under various anomaly rates and contamination parameters using semi-supervised learning with 10-fold CV.	166
Table 5.9 AUC performance comparison of various text representation methods across different anomaly rates and contamination parameters (ν), evaluated under a semi-supervised learning with 10-fold CV.	168
Table 5.10 Statistical comparison of OAE with transformer embeddings versus alternative text representations in semi-supervised learning, where all comparisons favour to the transformer-based text representation.	169

Table 5.11 Statistical comparison of the OAE against baseline models and other autoencoder variants in unsupervised learning settings.....	172
Table 5.12 Presents the AUC performance of baseline models versus the proposed autoencoder-based models, the models were tested under various anomaly rates and contamination parameters using unsupervised learning with 10-fold CV...	174
Table 5.13 Summarises the evaluation of semi-supervised and unsupervised models under similar settings (contamination rate of 0.05, anomaly level of 5%, 10-fold CV). The results presented that semi-supervised models achieved higher performance.	175
Table 6.1 The dataset size in English and Indonesian.....	190
Table 6.2 Bilingual dataset examples accompanied by their corresponding HS codes and extracted keywords.	191
Table 6.3 Hyperparameter configurations used in the experiments.....	193
Table 6.4 Complete 10-fold cross-validation (CV-10) test results of all models. The proposed models consistently outperform baseline models across evaluation metrics.....	195
Table 6.5 Statistical comparison between the proposed MK + DistilBERT model and various multilingual baseline models.	195
Table 6.6 Performance results from the ablation study, comparing the Products Only (PO) and Keywords Only (KO) models. The bottom section reports the performance gains of the proposed Mixed Keyword (MK) method over each variant.	199

List of Figures

Figure 2.1 Example of HS code structure applied to a laptop product category.	19
Figure 2.2 An example of a typical trade or customs declaration form.	20
Figure 2.3 Hierarchical structure with emphasis on leaf nodes for flat classification implementation.	28
Figure 2.4 Comparison between a hierarchical structure of a Tree and a DAG.	29
Figure 2.5 Types of classifiers used in hierarchical classification frameworks.	30
Figure 2.6 Monolingual embeddings before alignment (top) and after projection into a joint cross-lingual space (bottom). Source: (Ruder <i>et al.</i> , 2019).....	60
Figure 3.1 Overview of the classification pipeline comprising data pre-processing strategies, fine-tuning pretrained language models, and classification approaches.	71
Figure 3.2 Visualisation of the self-attention mechanism in a Transformer model using BertViz. This figure illustrates how the model dynamically allocates attention weights to different tokens, revealing the model's process of attending to other parts of the input sequence during the token encoding process.	78
Figure 3.3 Proposed flat classification network for trade product classification. This approach fine-tune pretrained Transformer-based models with a classification head composed of fully connected hidden layers on top of the encoder.	80
Figure 3.4 Proposed hierarchical network architecture for trade product classification. The classification system consists of three levels (Chapter HS-2, Heading HS-4, Sub-heading HS-6), each employing a transformer-based model as the base encoder. The network processes input and predicts labels at three hierarchical levels, with the output from each level passing to the subsequent level, enabling structured, level-wise classification.	83
Figure 3.5 10-fold cross-validation process for model evaluation, using rotating train-validation splits to ensure robust performance estimation.	85
Figure 3.6 Model performance of fine-tuned models on the trade dataset across different preprocessing strategies.	88

Figure 3.7 LIME feature visualisation of models using TF-IDF, Word2vec, ALBERT and DistilBERT to display the prediction probabilities assigned to a particular class based on the whole input text.....	91
Figure 3.8 Comparison of model performance versus training time using various transformer-based models with different preprocessing strategies with trade data. Fine-tuned DistilBERT model achieved both high performance and fast training time relative to other approaches.	96
Figure 4.1 Configuration of the proposed method using supervised contrastive learning to generate sentence embeddings for the downstream classification task.....	105
Figure 4.2 The pairwise sample construction process. Each anchor sample is pairing with positive samples, which are randomly selected from the dataset that share the same category (panel a), with the option of multiple positive samples allowed per anchor (panel b) to enable to learn from diverse product descriptions.....	108
Figure 4.3 Several methods configured to generate sentence embedding method to represent sentence-level embeddings for product descriptions.	118
Figure 4.4 Visualisation of frequency-based distribution of generated sentence embeddings from each method. Lighter colours indicate regions of higher similarity frequency among embeddings.	120
Figure 4.5 The average cosine similarity between the mean embedding vector and all individual embeddings in the dataset, accompanied by a visualisation of the similarity distribution. This analysis highlights the degree of anisotropy and embedding uniformity within the dataset.	121
Figure 4.6 The figure illustrates the distribution and clustering behaviour of embeddings across methods, highlighting the representational structure and separability of trade data within similar product categories in embedding space.	123
Figure 4.7 Duration of training time (hours: minutes: seconds) across two datasets using USE, SBERT, and the proposed method using SVM and RF classifiers (trained on an NVIDIA Tesla P100 GPU).....	128
Figure 4.8 Confusion matrix comparison between the fine-tuned DistilBERT model and the proposed method.....	128
Figure 5.1 A general overview of the proposed text-based anomaly detection framework for identifying misclassified trade transactions. The approach	

incorporates both semi-supervised and unsupervised learning methods to train the models. Anomalies are subsequently detected using a reconstruction-based mechanism, with threshold determination derived from the reconstruction error distribution.....	141
Figure 5.2 Integrated transformer-OAE architecture combining contextual sequence modelling with reconstruction-based anomaly detection.	145
Figure 5.3 Integrated transformer-VAE architecture combining contextual sequence modelling with probabilistic reconstruction for anomaly detection.	147
Figure 5.4 Integrated transformer-CAE architecture combining sequential modelling with convolutional reconstruction for anomaly detection.	149
Figure 5.5 Depicts the CAE encoder, where parallel convolutions with kernel sizes 3, 4, and 5 are combined with max pooling, concatenation, flattening, and dense layers to form the feature representation.....	150
Figure 5.6 Comparison tariff classifications and duty rates for selected products, including HS 84713010 (computers) and HS 85182100 (audio equipment). It presents that computers incur no import duty, while audio equipment is taxed at 10%.	153
Figure 5.7 Presents the ROC curve comparison of anomaly detection algorithms under varying contamination rates and anomaly levels. The OAE consistently outperformed all baseline methods and other autoencoder variants.	162
Figure 5.8 Visualisation of test data embeddings and corresponding latent representations using PCA for OAE, VAE, and CAE models (5% anomaly rate, contamination/nu = 0.05).....	164
Figure 5.9 OAE performance under semi-supervised learning (5% of anomaly rate and 0.05 of contamination/nu) presenting in scatter plots, loss densities, and confusion matrices.	170
Figure 5.10 CAE performance under semi-supervised learning (5% anomaly rate, contamination/nu = 0.05). The CAE generated greater reconstruction errors than the OAE, which resulted in false negatives and reduced anomaly sensitivity...	171
Figure 6.1 Overview of the methodology for text and mixed keyword integration in bilingual product classification, consisting of: a keyword prediction model, sample configuration, and a bilingual classification model for predicting product categories.....	184
Figure 6.2 Distribution of bilingual data across product categories.	190

Figure 6.3 Embedding space visualisation of Indonesian (ID) and English (EN) trade data using PCA. The left panel represents embeddings produced by the fine-tuned multilingual XLM-R model and the right panel shows our proposed MK method.	196
Figure 6.4 Heatmap showing similarity scores between inter-class and intra-class labels across Indonesian (y-axis) and English (x-axis) embeddings, generated by the fine-tuned XLM-R and the proposed method. The diagonal represents identical class labels across both languages. Darker colours indicate higher similarity, with dark blue regions highlighting strong alignment between the same class labels in the two languages.	197
Figure 6.5 Retrieved product descriptions using bilingual queries. Despite variations in the English and Indonesian input terms, the model consistently retrieves products with matching HS codes, demonstrating its effectiveness in processing bilingual queries and aligning semantically equivalent descriptions.	198
Figure 6.6 Performance of the bilingual model trained with a gradually increasing number of Indonesian training samples. ALL indicates the model's performance on the full bilingual test set, EN represents performance on English test data, and ID reflects performance on Indonesian test data.	200

List of Abbreviations

AHTN	ASEAN Harmonised Tariff Nomenclature
AI	Artificial Intelligence
ALBERT	A Lite BERT
ARCCA	Advanced Research Computing at Cardiff
AUC	Area Under the Curve
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long Short-Term Memory
BoW	Bag-of-Words
BPE	Byte Pair Encoding
CAE	Convolutional Autoencoder
CBOW	Continuous Bag-of-Words
CBP	Customs and Border Protection
CN	Combined Nomenclature
CNN	Convolutional Neural Network
CV	Cross-Validation
DAG	Directed Acyclic Graph
DAN	Deep Averaging Network
DeBERTa	Decoding-enhanced BERT with Disentangled Attention
DSS	Decision Support System
DT	Decision Trees
ELECTRA	Efficiently Learning an Encoder That Classifies Token Replacements
ELMo	Embeddings from Language Models
FNN	Feedforward Neural Network
FPR	False Positive Rate
GloVe	Global Vectors
GPT	Generative Pretrained Transformer
GPU	Graphics Processing Unit
GRI	General Rules for Interpretation
GRU	Gated Recurrent Unit
HS	Harmonized System
HTS	Harmonized Tariff Schedule
IFO	Isolation Forest
KO	Keywords Only
LDA	Latent Dirichlet Allocation
LOF	Local Outlier Factor
LSTM	Long Short-Term Memory

MAE	Mean Absolute Error
MK	Mixed Keywords
MLM	Masked Language Model
MLP	Multilayer Perceptron
MSE	Mean Squared Error
MUSE	Multilingual Universal Sentence Encoder
NCM	Nomenclatura Común del Mercosur
NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
NPLM	Neural Probabilistic Language Model
NSP	Next Sentence Prediction
NTP	Next-Token Prediction
OAE	Original/Standard Autoencoder
OC-SVM	One-Class Support Vector Machines
OHE	One Hot Encoding
OOV	Out-of-Vocabulary
PCA	Principal Component Analysis
PO	Products Only
RF	Random Forests
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RoBERTa	Robustly Optimized BERT Pretraining Approach
ROC	Receiver Operating Characteristic
RTD	Replaced Token Detection
SBERT	Sentence-BERT
Seq2Seq	Sequence to Sequence
SOP	Sentence Order Prediction
SVM	Support Vector Machines
TF-IDF	Term Frequency-Inverse Document Frequency
TPR	True Positive Rate
USE	Universal Sentence Encoder
VAE	Variational Autoencoder
WCO	World Customs Organization
Word2Vec	Word to Vectors
WTO	World Trade Organization

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors, Dr Yuhua Li, Dr Padraig Corcoran, and Dr Dennis de Widt, for their guidance and support throughout the completion of this thesis. I am deeply grateful to Dr Yuhua for allowing me to pursue my PhD under your supervision. Your unwavering support, understanding, remarkable kindness, and valuable research guidance have been crucial to the completion of this thesis. I am sincerely grateful to Dr Padraig for his invaluable expertise and unwavering support, especially during challenging times, which helped guide me throughout my studies. Finally, I would like to thank Dr Dennis de Widt for his support and involvement in the initial stages of my study.

I am immensely thankful to my lovely wife, Ani, for her unwavering patience and continual support. Our wonderful moments, your unconditional love, and encouragement have been my anchor, guiding me through the challenges of this journey. I also wish to thank my daughters, Jasmine and Keenar, whose love, joy, and even occasional challenges have brought balance to my life and inspiration throughout this journey. I owe my deepest thanks to my parents and parents-in-law for their unconditional love, support, and prayers, which have sustained me throughout every stage of this study. I am grateful to my colleagues at Abacws for making these four years an enjoyable experience, and to my Indonesian friends for their support and companionship in both study and social life.

I extend my sincere gratitude to my supervisors and colleagues at the Directorate General of Taxes for their guidance and for providing the necessary resources throughout my PhD journey. Finally, I extend my sincere thanks to the Indonesia Endowment Fund for Education (LPDP) for generously awarding me a scholarship to pursue my PhD.

Chapter 1

Introduction

1.1 International Trade

International trade is defined as a cross-border exchange of products and services (Kim, 2021). It has been recognised as a key driver for economic development, exhibiting the magnitude in the worldwide trade flows, which surpassed 40 trillion USD, combining export and import values¹. The trend of global trade is expected to continue, with a 3.3% growth in trade volume projected by 2025 (World Trade Organization, 2024). The revenue generated from trade activities is expected to increase proportionally with trade volumes, contributing to national revenue and supporting national economic development. On average, tariffs on imported products constitute more than 30% of national tax revenue (World Customs Organization, 2024a). Besides serving as a national income source, governments use tariffs to protect domestic interests and influence trade flows. These tariffs are selectively applied to certain countries or products to make domestic products more price-competitive and encourage local purchasing, boosting economic activity (Azzimonti *et al.*, 2025).

The international trade procedure generally necessitates the utilisation of documents for financial, transport, and administrative purposes (Kim, 2021). Among these, the trade or customs declaration is a key source document. This document functions as a fundamental instrument in international trade, forming part of the standardised documentation framework that both regulates and facilitates the cross-border exchange of products (World Customs Organization, 2017). These documents are required for customs clearance to ensure procedural validity and compliance before

¹ <https://comtradeplus.un.org/TradeFlow>

products are authorised to enter or exit a country (Gani, 2016).

International trade is recognised as a domain that generates vast amounts of data, particularly text data, since many elements of trade documents are expressed in free-text format, such as product descriptions (Matsudaira and Koh, 2022). Over the past few years, the submission of trade declarations has substantially increased globally from 601 million in 2018 to around 1,089 million in 2024 (World Customs Organization, 2018a, 2024a). This increase in document submissions corresponds to the increase in volume of international trade. As a consequence, managing and processing the volume of submissions on a daily basis presents substantial challenges.

Accordingly, trade facilitation should ensure that stakeholders in international trade are provided with the necessary facilities and support to enable the smooth execution of trade processes. Traders depend on complete and accurate information in the trade declaration for document submission. From the perspective of trade authorities, effective and comprehensive validation of trade declarations is important to guarantee adherence with regulatory requirements and optimal collection of national revenue. This interdependence is important for traders and authorities in achieving valid and reliable trade processes.

Digital transformation has substantially reshaped operational models and value across industries, including public sector organisations (Shakil and Tasnia, 2022). Tax and customs administrations, which are part of this of digital reformation, are transitioning from basic digitisation towards full digitalisation. However, despite advances in modern information and communication technology, some administrations continue to struggle with low performance in meeting demands for faster and more reliable public services (Matsudaira and Koh, 2022). The increase in the volume of global trade amplifies the necessity to develop advanced solutions.

Emerging technologies have revealed potential to support regulatory authorities, especially in tackling issues in tax and customs administration (Heijmann *et al.*, 2020; Rukanova *et al.*, 2021). One example is manifested in the adoption of automated systems and electronic platforms designed to accelerate trade and customs clearance processes by providing traders with tools to process trade documentation (Heijde, 2019; Lux and Matt, 2021), which in turn improves the effectiveness of international

trade systems (World Customs Organization, 2017). Additionally, using advanced data analytics, like predictive models for trade data, can reveal patterns in complex tasks and improve trade facilitation (Banker, 2017; Rukanova *et al.*, 2021). At the same time, implementing such technological breakthroughs encourages the transformation of underutilised trade data into actionable knowledge, supporting better decision-making and improving the efficacy, validity, and efficiency of trade operations.

1.2 Challenges of Automating Tasks in International Trade

Digital transformation and technological implementation in international trade have not fully addressed persistent challenges and have yet to cover critical processes adequately. These include the accuracy of product classification, product misclassifications or anomalies within trade declarations, and multilingual product descriptions. These challenges constrain trade effectiveness and regulatory compliance, thus necessitating automated solutions. Each challenge is examined in detail below.

1.2.1 Trade Product Classification

The trade product should be classified to a specific product code to determine applicable tariff rates for the associated product, facilitate permit approval, and support statistical reporting purposes (Asakura, 1993). The product classification system, known as the Harmonized System (HS), was developed by the World Customs Organization (WCO) and provides a global framework for product classification in international trade (World Customs Organization, 2018b).

However, the activity of product classification involving matching product descriptions with their valid HS codes remains challenging (Brigstock, 2017; Grainger, 2024; Heijde, 2019). This recurring challenge in product classification originates from several factors, including the inherent complexity of HS and gaps in terminology used in industry and in the HS guidance (Heijde, 2019). The HS is composed of thousands of product categories, which resemble a hierarchical structure consisting of complex distinctions within each group, which often create ambiguous interpretations between categories. Consequently, a single product may reasonably fall under multiple categories. Terminological gaps in the guidelines also pose complexities, because industrial or commercial terminologies frequently differ from the official descriptions in

the HS guidance (Kappler, 2011). As a result, it is often assumed that the assignment of HS codes can only be reliably performed by an expert with sufficient experience (Edirisinghe, 2017). Ongoing WCO initiatives aim to enhance the HS by identifying areas for classification ambiguities and making the classification process more transparent (Christopher Starns, 2024; World Customs Organization, 2024b).

Consequently, the provision of a reliable assistive tool for HS code classification is considered important to mitigate the risks associated with different interpretations in product classification (Grainger, 2024). The availability of such automated instrument can support to avoid consequences that can be severe, ranging from substantial fines to the seizure of products and delays in customs clearance.

1.2.2 Product Misclassification

Another major challenge in international trade concerns product misclassification, which occurs when a product is assigned to the wrong product category. A study demonstrated that while some misclassification is unintentional, it was frequently represented as a deliberate method of customs duty and tax evasion (Working Group on Commercial Fraud, 2010). The consequences of product misclassification are harmful to national interests. The national revenue collection can be disrupted by the miscalculation of duty and tax rates, which is caused by the intentional assignment of products to lower-tax categories (Constantine Cannon, 2020; Lushi and Gashi, 2016). The financial consequences of product misclassification have been documented in several jurisdictions (Basalisco, 2016; Ding *et al.*, 2015; European Court of Auditors, 2017; Office of The Auditor General of Canada, 2017). In addition, inaccurate classification can also compromise the integrity of trade statistics, as it is necessary for analysing decision-making regarding trade policy. Ignoring this issue will continuously weaken the effectiveness of the authority in justifying evidence-based policymaking.

The issue of product misclassification requires special attention from tax and customs authorities, given that these anomalies are challenging to identify and require practical tools for trade authorities to monitor and detect such anomalies (Chalendard, 2017). Providing an automated monitoring and validation system for trade authorities can potentially mitigate these challenges by deterring non-compliance while improving the

detection and mitigation of potential tax and customs violations in international trade.

1.2.3 Bilingual Languages Classification

The presence of bilingual languages in international trade data can be attributed to trade declaration submissions from traders with diverse linguistic backgrounds (Paramartha *et al.*, 2021). The use of a free-text field allows unlimited variation in describing product information, creating subsequent complexity, for example, the text contains bilingual languages that complicate data processing and validation. On the other hand, some regulatory frameworks allow international language input to improve the completeness and accuracy of product descriptions (Indonesian Customs, 2009). For trade authorities, this issue causes an operational workload because it requires officials to comprehend both product classifications and terminology outside the domestic language. On the other hand, traders with varied backgrounds expect to submit products without causing any disruption in the submission process.

Providing an automatic tool that functions in the presence of multiple languages requires more than an understanding of the technical structure of the HS; it also requires an understanding of equivalent product terminology expressed in different languages. To support traders in accurately classifying products into valid categories, a reliable tool equipped with features that address language barriers in global trade is important (Grainger, 2024).

1.3 Research Landscape in International Trade

In the last decade, studies on automated solutions for international trade have increased, mainly due to developments in machine learning and natural language processing (NLP). This section provides an overview of selected research, focusing on the limitations of existing studies related to the identified challenges in international trade. These limitations establish the basis for the aims and objectives of this study.

1.3.1 Text Classification Task

Text preprocessing is the first step usually used in the text classification pipeline (Kowsari *et al.*, 2019), whose role is to transform the raw text into a formatted form that can be effectively processed in the next steps. Therefore, selecting appropriate preprocessing is important for text-based tasks because the input quality has a strong

influence on classification performance (Uysal and Gunal, 2014). However, this process has not been sufficiently addressed in the study of international trade, where the studies typically apply standard techniques without a systematic evaluation of the effectiveness using trade data. Text preprocessing typically involves transforming the text input by preserving or removing specific elements or product attributes. It is considered an important step, especially when applied to process domain-specific datasets, such as trade data, which consist of mixed and heterogeneous attributes that are expected to affect model performance. However, the preprocessing step is often overlooked, which is amplified in the case of deep learning implementation (Camacho-Collados and Pilehvar, 2018). A research study found that text preprocessing influences the model performance (Siino *et al.*, 2024), including the ones that utilised modern language models.

Research on product classification in the trade domain has explored diverse text representation methods, ranging from traditional encodings (e.g., TF-IDF, Bag-of-Words) to modern language models, such as transformer-based architecture (Vaswani *et al.*, 2017). However, little research has systematically compared these approaches in a specific domain such as trade data. Modern pretrained language models have achieved state-of-the-art performance in many NLP-based tasks (Li, Peng, *et al.*, 2022), but it is important to identify which language models are most effective for product classification. Multiple factors contribute to determining the most effective models, given the large number of publicly available models and the computational resources needed for their implementation (Wolf *et al.*, 2020). Additionally, these models, however, have been pretrained on general-purpose corpora whose linguistic and structural characteristics differ substantially from trade data, which is typically short and lacks grammatical structure.

Product classification tasks have investigated various classification approaches that have been adopted in prior studies, each justified by specific performance optimisation goals. Most research has relied on a flat classification approach rather than hierarchical classification approaches, despite HS codes resembling a hierarchical structure (Silla and Freitas, 2011), leaving important questions unresolved. Therefore, developing hierarchical classification and performing a comparative study between flat and hierarchical methods is required to establish which approach contributes to the

improvement of model performance.

1.3.2 Sentence Representation Learning

Beyond word-level representations, sentence-level representations provide an essential alternative input for machine learning classifiers, which enhance text quality and have a significant impact on the performance of classification tasks (Conneau and Kiela, 2018; Ramesh Kashyap *et al.*, 2024). In the trade domain, sentence-level representations have been applied in limited studies for HS code classification, where most of the studies relied on off-the-shelf models without further fine-tuning, a practice regarded as suboptimal given trade-specific linguistic patterns and terminology (Cer *et al.*, 2018; Pain, 2021). The limitation is further increased by the tendency of pretrained transformer-based language models to generate highly similar text representations (Ethayarajh, 2019), which reduces their ability to distinguish between closely related product categories.

Contrastive learning has been proposed as a method to enhance the capacity of text features, in particular those generated from pretrained language models (Jaiswal *et al.*, 2021; Le-Khac *et al.*, 2020; Xu, Xie, *et al.*, 2023). Although this method is considered effective, the use of this approach in domain-specific settings, especially international trade, is underexplored. This gap motivates this study to explore its implementation, to perform representation learning and to improve text classification performance, producing standardised representations of trade product declarations for efficient processing in classification tasks.

1.3.3 Text Anomaly Detection

Previous studies on anomaly detection in the trade domain have utilised heterogeneous data types, including images and tabular data, which were predominantly used to address tax and customs evasion (Abdolshah *et al.*, 2017; Kim *et al.*, 2020). On the other hand, academic work utilising text data in the area of international trade is still limited, whereas recent breakthroughs in processing and understanding textual data offer a promising approach for developing anomaly detection tasks. A further limitation in the existing literature is that studies rely on supervised learning, which assumes the availability of labelled datasets (Chan *et al.*, 2023). However, in practice, such labels are scarce as they are sensitive and

challenging to obtain because of the specific method used by the jurisdiction in employing anomaly detection methods.

Detecting anomalies in text data, which can be applied for detecting product misclassification, remains an underexplored research area (Novoa-Paradela *et al.*, 2024; Xu, Gábor, *et al.*, 2023). Previous studies using supervised learning are likely to achieve suboptimal performance in detecting misclassified transactions, since these methods have difficulty capturing evolving anomalous patterns. In a review study of anomaly detection utilising various deep learning approaches (Ruff *et al.*, 2021), reconstruction-based methods achieved the best performance compared to others, such as probabilistic and one-classification methods. However, this empirical finding was based only on image-type data. Building on this finding, combining reconstruction-based methods with modern language models becomes a promising approach for detecting anomalies. Autoencoders are a class of neural network architectures that were originally used for dimensionality reduction but have since been recognised for their capability in anomaly detection (Berahmand *et al.*, 2024; Hawkins *et al.*, 2002; Japkowicz *et al.*, 1995). This architecture has proven effective for anomaly detection in various studies, but its use in text-based NLP tasks is still limited (Li, Pei, *et al.*, 2023). Recognising this gap, autoencoder networks present a viable method for detecting product misclassifications in international trade, since they are capable of working in unsupervised settings and can process varied data types (Pang *et al.*, 2022).

1.3.4 Bilingual Classification Task

Most studies on trade product classification have been performed in monolingual contexts, meaning that the model is intended to cope with single-language trade data (Luppès, 2019; Ruder, 2020; Shubham *et al.*, 2023; Spichakova and Haav, 2020a). Compared to other languages, English datasets are available in the public domain and have become the motivation for several studies to develop models for comparison and improvement. On the other hand, non-English literature on trade product classification usually involves cooperation with specific country jurisdictions, where the generated datasets are generally not publicly available because of non-disclosure agreements and the confidential nature of trade data (Lee *et al.*, 2024; Paramartha *et al.*, 2021). The literature indicates that trade transactions in the real world often include bilingual

text (Marra de Artiñano *et al.*, 2023; Paramartha *et al.*, 2021). On the other hand, there is a lack of research on multilingual product classification in trade, which limits the development of automation systems that can manage the linguistic diversity in global commerce.

Modern language models, especially multilingual models, are considered a promising solution to address the issue of bilingual trade product classification since they have been trained on diverse language corpora that enable multilingual understanding (Feijo and Moreira, 2020). The implementation of these models is also based on the assumption that they produce aligned representations across languages to address multilingual challenges (Bhargava *et al.*, 2023). The development of a multilingual model for product classification is not feasible because of the limited multilingual trade data in the public domain. Additionally, the strong assumption about multilingual model capability conflicts with the finding that utilising multilingual models achieved lower performance than language-specific models on monolingual data (de Lima *et al.*, 2022), showing that they have limitations compared to monolingual models. A feature engineering approach represents a promising method to enrich text features by employing data augmentation techniques, and previous studies have explored the use of keyword-based features (Nair *et al.*, 2024; Xiang *et al.*, 2020). The integration of these methods with modern language models shows potential for addressing the challenges of bilingual product classification, particularly in low-resource languages.

1.4 Research Aims, Questions, and Objectives

The overarching aim of this thesis is to develop predictive models that have high performance and potential practical applicability in the trade domain. To achieve this purpose, this thesis leverages modern pretrained language models, such as transformer-based models, and the implementation of deep neural networks to address the challenges in product classification, representation of product data, product misclassification, and bilingual classification.

The proposed models are designed to function as decision-support tools within customs and administrative workflows. Accordingly, the approach is intended to complement, rather than replace, human oversight. Final review, validation, and decision-making remain the responsibility of end-users, thereby preserving

accountability and professional judgment.

In this context, the development of classification models raises important questions regarding model architecture, representation learning strategies, and classification approach using real-world trade data. The challenge involves not only enhancing classification accuracy but also ensuring robust evaluation, model interpretability, and feasibility of deployment within operational environments. Accordingly, this research is guided by the following overarching research question (RQ):

How can modern language models and deep neural networks be developed to provide accurate and practical solutions for international trade product classification systems?

The study addresses the main research question through the following sub-questions:

- RQ1: How to develop a classification task that involves language models with a domain-specific dataset that performs well in classifying international trade products into HS codes?

To address this question, the thesis aims to propose a text classification pipeline for product categorisation in the trade domain, which includes applying preprocessing strategies, fine-tuning modern language models, and developing classification approaches.

- RQ2: How can the representation of product descriptions be leveraged to improve discrimination between closely related product categories and improve classification performance?

This research responds to the question by proposing a method to generate sentence-level representation utilising supervised contrastive learning for product declaration to improve product classification performance and develop a flexible representation to be used by machine learning classifiers.

- RQ3: How can text anomaly detection methods be applied with product classification models to identify potential misclassifications?

The investigation of this question is undertaken through proposing a method to identify product misclassification using text anomaly detection that combines

autoencoder architectures with modern language models, applying semi-supervised and unsupervised learning techniques.

- RQ4: How to improve classification performance for bilingual product declarations in the context of low-resource international trade data?

This objective is pursued through proposing a model for bilingual product classification, which involves feature engineering of keyword-based features, pretrained language models, and deep learning architectures to enhance alignment representation of different product descriptions across languages to improve classification accuracy, in particular in low-resource trade settings.

1.5 Research Contribution

The contributions of this thesis to machine learning in international trade span four areas: automation of product classification, enhanced product representation, detection of product misclassifications, and application of bilingual classification. The main contributions of this thesis are as follows:

1. **The development of a complete pipeline for trade product classification.** The classification pipeline consists of three primary components, which start from employing a set of text preprocessing strategies designed to preprocess textual trade data, providing insight into the impact of these strategies on the model performance. An understanding of modern language models is obtained, including key metrics consisting of training efficiency and model performance by fine-tuning pretrained language models using trade datasets. A key innovation introduced in this thesis is the implementation of a hierarchical classification method, which significantly outperforms the flat classification that has become a predominant approach frequently utilised in the literature in trade. Finally, the end-to-end implementation of the proposed classification pipeline produces an improvement in product classification performance with an empirical contribution of more than a 22% gain in accuracy compared to an existing study, evaluated on publicly available real-world datasets containing over 3,000 product categories.
2. **The generation of product descriptions through enhanced sentence-level representation.** The primary key proposed in this development of a

representation learning is the involvement of a supervised contrastive learning framework, which encourages the model to generate representations that bring similar products closer together while pushing apart those from different categories. By leveraging modern language models and a shared network architecture, this method generates enhanced sentence-level representations for product descriptions, and it mitigates the issues commonly found in generated transformer-based representations. The empirical contribution shows that the enhanced sentence-embeddings combined with traditional classifiers outperform fine-tuning methods in trade product classification.

3. **The development of a text anomaly detection model capable of identifying product misclassification in trade datasets.** The proposed method employs a range of autoencoder architectures to learn latent representations of normal trade data patterns. Language models are employed to generate rich text representations that enhance the autoencoder's ability to learn from contextualised representations of text data, which subsequently are used for autoencoder networks in both semi-supervised and unsupervised scenarios for text anomaly detection. Empirical results show that the proposed models effectively detect anomalies across different anomalous levels and outperform well-established anomaly detection baselines. The findings reveal that the proposed simple variants of autoencoders achieve the best performance in both semi-supervised and unsupervised learning, which practically provides the ability to detect text anomalies manifesting in product misclassifications in the trade domain.
4. **Proposes an innovative method for bilingual product classification to address the presence of bilingual languages in the trade data.** The proposed method utilises a feature engineering method that aims to combine the text features with keyword-based features derived from product descriptions. A language model is used to generate a richness of the combined features to train the model through a deep learning architecture designed for bilingual classification. Experimental evaluation utilising English and Indonesian datasets presents that the proposed method significantly outperforms baseline methods that rely on fine-tuned multilingual models and contributes to the alignment of trade

product representations across languages, reducing dependency on low-resource languages.

A summarised list of all contributions that have already been published, submitted, and prepared for publication is as follows:

- Chapter 3 of the thesis is based on the following accepted paper: **Anggoro, A.**, Corcoran, P., De Widt, D., and Li, Y. (2023). Using Distilbert to Assign HS Codes to International Trading Transactions. *World Conference on Information Systems and Technologies*, Pisa, Italy, 4 - 6 April 2023. AA developed the methodology and carried out the analysis with input from YL as the main supervisor. AA led the writing of the paper, with feedback from YL, PC, and DDW. The paper has been substantially improved and rewritten by AA for its inclusion as a chapter in the thesis, which represents AA's own work.
- Chapter 4 of the thesis is based on the following accepted paper: **Anggoro, A.**, Corcoran, P., De Widt, D., and Li, Y. (2025). Harmonized system code classification using supervised contrastive learning with sentence BERT and multiple negative ranking loss. *Data Technologies and Applications*, 59 (2), pp. 279-301. 10.1108/DTA-01-2024-0052. AA developed the methodology and carried out the analysis with input from YL and PC. AA led the writing of the paper, with feedback from YL, PC, and DDW, and advice from reviewers. The paper has been substantially enhanced and rewritten by AA for its inclusion as a chapter in the thesis, which represents AA's own work.
- Chapter 6 of the thesis is based on a submitted paper for publication: **Anggoro, A.**, Corcoran, P., and Li, Y. (2025). Enhancing Cross-Lingual Embedding Alignment with Additive Keywords for International Trade Product Classification. This paper has been submitted and is currently under review. AA developed the methodology and carried out the analysis, with the assistance of YL and PC. AA led the writing of the paper, with input from YL and PC. AA has rewritten the paper for inclusion as a chapter in the thesis, which represents a more comprehensive work of AA's own research compared to the submission, given the limited page allowance.
- Chapter 5 of the thesis is based on unsubmitted work: **Anggoro, A.**, Corcoran, P., and Li, Y. Text Anomaly Detection for Trade Product Misclassification via Autoencoders. AA developed the methodology, carried out the experiments, and

analysed the data with input from YL and PC. AA led the writing of the paper, with input from YL and PC. The paper has been rewritten by AA and converted into a manuscript style with publishable quality.

1.6 Thesis Structure

This thesis is organised into seven chapters, outlined below:

Chapter 2 reviews the literature on machine learning tasks in the international trade domain. The section begins by introducing the HS codification framework utilised in international trade, describing the product code structure, illustrating an example of a trade document and summarising trade datasets used in prior studies. The next chapter reviews the literature organised thematically, starting from text classification, representation learning, anomaly detection, and bilingual classification. Each theme is discussed alongside its relevant theoretical background, and the chapter concludes by identifying gaps that motivate the proposal of methods in this thesis.

Chapter 3 presents the methodological framework developed for classifying trade products. The method sets text preprocessing strategies and continues to fine-tune pretrained language models to generate text representations. Both flat and hierarchical classification frameworks are then introduced for product categorisation. The chapter presents a comparative study between baseline methods and the proposed approach to demonstrate the model's effectiveness and performance in product classification.

Chapter 4 presents the development of a predictive model for classifying trade products based on sentence-level representations. Central to this chapter is the introduction of supervised contrastive learning, utilising a Siamese network architecture, as presented in the methodology section. The chapter also describes the construction of paired training samples and the configuration of the contrastive learning framework, followed by details on applying the resulting representations to classification tasks. The result and discussion section presents the findings of the evaluation of the proposed methods, which are divided into two: an assessment of sentence representation quality and a comparative analysis against baseline models. The next explanation provides the effects of hyperparameter optimisation on model performance before the chapter concludes with the summary section.

Chapter 5 presents an approach to developing a text anomaly detection model for identifying misclassified trade transactions. The chapter begins with presenting some literature that motivates the research. In the later subsection, this chapter provides an overview of the proposed method, introducing the mechanism of autoencoder networks in learning data patterns and the configuration of the networks used in the experiment. The experimental results are presented, along with a discussion of the proposed models' findings compared to established baseline methods in various learning settings. The summary section concludes the chapter.

Chapter 6 presents the development of a classification model used to categorise bilingual product presence in trade data. The earlier section in this chapter provides brief information about multilingual classifications in the literature. The discussion is continued by presenting the methodology used to model the bilingual model, which consists of a method for developing keyword extraction and a keyword classifier. This section provides an overview of the dataset used and the evaluation metrics, which include three key aspects: performance compared to baseline models, cross-lingual representational alignment, and effectiveness in low-resource language contexts. The results and discussion of the empirical findings are presented before the summary of the study.

Chapter 7 presents the conclusion of this thesis. It contains a summary of the thesis's key findings and contributions, along with potential avenues for future research in machine learning applications for international trade.

Chapter 2

Literature Review: Foundations of Automated International Trade Classification Systems

This review provides an overview of the relevant literature on the application of machine learning methods to problems in the domain of international trade. The review adopts a structured narrative approach to explain and critically discuss previous studies, identifies key themes, debates, gaps, and trace the development and underlying theoretical foundations within the field. It examines prior research focuses on peer-reviewed journal articles, conference proceedings, books and technical reports published within the fields of machine learning and NLP. Literature was identified through targeted searches of major academic databases using keywords related to HS code classification, trade product classification, sentence-based classification, transformer-based classification models, anomaly detection for trade data, and multilingual or bilingual text classification. The selection prioritised studies that proposed, evaluated, or reviewed computational methods applicable to real-world trade or customs contexts.

The chapter begins with an introduction to international codification systems and associated documentation used for classifying traded products, followed by a summary of datasets employed in research across a range of tasks within international trade. The remaining sections review research studies organised into four thematic areas: text classification methods covering both word-level and sentence-level representations, text anomaly detection approaches in the trade domain, bilingual product classification studies, and relevant theoretical background for each area. Each section will be concluded with a summary of the identified research gaps, which

subsequently form the foundation of this research objective, leveraging pretrained language models and neural network architectures for task-specific applications in the international trade domain.

2.1 International Trade Data

The section introduces a codification system used widely in the international trade domain and underlines its usage along with its challenges. A trade or a customs declaration is presented as a document that is central to this thesis, and a summary study associated with datasets that have been utilised in the international trade study.

2.1.1 Harmonized Systems

WCO developed a standardised international product nomenclature known as the Harmonized Commodity Description and Coding System, commonly referred to as the Harmonized System (HS) in 1988 (World Customs Organization, 2018b). The HS is implemented by customs authorities in international trade as guidance to classify trade product categories and is used as a basis to set tariffs. The system is organised in a logical and hierarchical structure that provides the uniform classification of traded products to cover over 98% of products in international trade (Edirisinghe, 2017; World Customs Organization, 2018b).

The HS structure consists of groups of products organised and sorted by the degree of technological complexity, spanning from raw materials to manufactured goods. The HS is divided into sections, which are the highest-level grouping in the HS code hierarchy. However, they are not part of the actual HS code number itself. Each section groups related products into one of 21 major categories, beginning from basic commodities, such as Section I: Live Animals and Animal Products, Section II: Vegetable Products, and progressing to more advanced technological products, such as Section XVI: Machinery and Mechanical Appliances and Electrical Equipment. Sections contain multiple chapters, and each chapter represents the first two digits of the HS code, which describes broad categories of products. In each chapter, more specific classifications are categorized by four-digit headings and further detailed into six-digit subheadings, facilitating more precise distinctions among similar products, as shown in Figure 2.1. The HS classification system consists of 1,228 headings and 5,612 subheadings, accommodating a mechanism for classifying a wide range of trade

products (World Customs Organization, 2018b), as summarised in Table 2.1.

HS Code Type	Description	Number of Categories
-	Section	21
2-digit HS codes (HS-2)	Chapter	97
4-digit HS codes (HS-4)	Heading	1,228
6-digit HS codes (HS-6)	Sub-heading	5,612

Table 2.1 Hierarchical structure of HS codes based on digit level.

The six-digit HS code is a globally mandatory code that countries must implement to facilitate the exchange of trade products across borders. However, each country can extend the standard six-digit HS code by adding additional digits for national tariff and statistical purposes. The HS, which generally consists of the product code, categories names, and tariff, can specifically be implemented for a country, such as the Harmonized Tariff Schedule (HTS)² used in the the United States, or for a region, such as Combined Nomenclature (CN)³ adopted within the European Union, the ASEAN Harmonised Tariff Nomenclature (AHTN)⁴ for Southeast Asia and Nomenclatura Común del Mercosur (NCM)⁵ applied in South American countries.

² <https://hts.usitc.gov/>

³ <https://trade.ec.europa.eu/access-to-markets/en/content/combined-nomenclature-0>

⁴ <https://asean.org/harmonisation-of-tariff-nomenclature-customs-valuation-and-procedures/>

⁵ <https://www.mercosur.int/politica-comercial/ncm/>

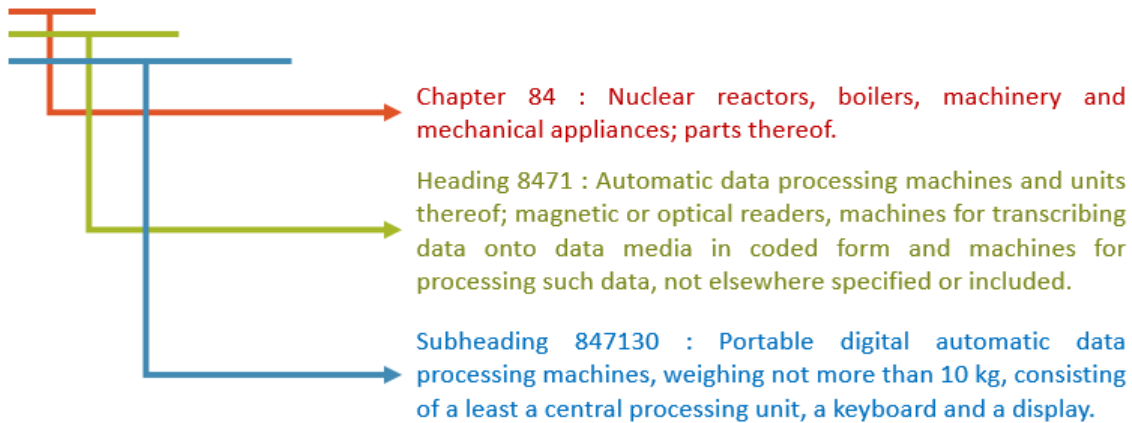
847130

Figure 2.1 Example of HS code structure applied to a laptop product category.

2.1.2 Trade Declaration

A trade declaration, also known as a customs declaration (see Figure 2.2), is a legally required document that provides details about products for exchange across international borders. The document is basically a formal notification to trade authorities, detailing the product description, quantity, value, origin, destination, and HS codes. The process of assigning HS codes to product descriptions is known as HS classification, which is important for traders to understand and execute correctly because the accuracy of classification will be validated in the assessment of duties, taxes, and regulatory compliance. The assigned product codes are then used to reference HS guidance tables, which contain the information needed to assess duties and taxes, enforce trade regulations, and ensure compliance with applicable trade policies. To support traders in assigning valid product codes, a set of principles known as the General Rules for Interpretation (GRI) is utilised to provide guidance for classification decisions (World Customs Organization, 2022). The determination of a product code depends on various factors related to the characteristics of the products.

Great Britain Grande-Bretagne		CUSTOMS DECLARATION				CN 23	
From De	Name	Sender's Customs reference (if any) Référence en douane de l'expéditeur (si elle existe)		No. of item (barcode, if any) May be opened officially		Important! See instructions on the back	
	Business			DÉCLARATION EN DOUANE			
	Street			N° de l'envoi (code à barres, s'il existe) Peut être ouvert d'office			
	Postcode						
	Country						
To À	Name			Importer's reference (if any) (tax code/VAT No./importer code) (optional) Référence de l'importateur (si elle existe) (code fiscal/N° de TVA/code de l'importateur) (facultatif)			
	Business						
	Street			Importer's telephone/fax/e-mail (if known) N° de téléphone/fax/e-mail de l'importateur (si connus)			
	Postcode						
	Country						
Detailed description of contents (1) Description détaillée du contenu		Quantity (2) Quantité	Net Weight (3) Poids Net (in kg)	Value (5) Valeur	<i>For commercial items only</i> <i>Pour les envois commerciaux seulement</i> HS tariff number (7) N° tarifaire du SH		
Product description					Country of origin of goods (8) Pays d'origine des marchandises		
					HS codes		
			Total gross weight (4) Poids brut total	Total value (6) Valeur totale	Postal charges/Fees (9) Frais de port/Frais		
Category of item (10) Catégorie de l'envoi ☐ Gift Cadeau ☐ Documents		Commercial sample Echantillon commercial ☐ Returned goods Retour de marchandise ☐ Other Autre		Explanation: Explication:		Office of origin/Date of posting Bureau d'origine/Date de dépôt	
Comments (11): (e.g.: goods subject to quarantine, sanitary/phytosanitary inspection or other restrictions) Observations: (p. ex. Marchandise soumise à la quarantaine/à des contrôles sanitaires, phytosanitaires ou à d'autres restrictions)						I certify that the particulars given in this customs declaration are correct and that this item does not contain any dangerous article or articles prohibited by legislation or by postal or customs regulations	
☐ Licence (12) Licence No(s). of licence(s)		☐ Certificate (13) Certificat No(s). of certificate(s)		☐ Invoice (14) Facture No. of invoice		Date and sender's signature (15)	

Figure 2.2 An example of a typical trade or customs declaration form⁶.

2.1.3 Trade Datasets

Several studies in the domain of international trade utilised datasets that were derived from customs jurisdictions and logistics providers. However, as illustrated in Table 2.2, most of these datasets are not publicly available. Only a limited number of datasets are in public domain. These datasets are often subject to non-disclosure agreements with data providers because of the sensitive nature of the information. In terms of the classification scope of product categories or HS codes, only a limited number of studies employed the full range of HS codes, as presented in Table 2.2.

⁶ <https://www.royalmail.com/>

Data Source	Publicly Available	Study	HS codes in the study
U.S. import data – Customs and Border Protection	Yes ⁷	(Ruder, 2020) (Spichakova and Haav, 2020a) (Luppess, 2019) (Shubham <i>et al.</i> , 2023)	All HS-6: 3,243 classes All HS-6: undefined classes All HS-6: HS-2: 96, HS-4: 1207 Chapter 84
India import data	Yes ⁸	(Harsani <i>et al.</i> , 2020)	Chapter 64
Anonymous Transport and Shipping Company	No	(Pain, 2021)	-
Singapore Customs	No	(Ding <i>et al.</i> , 2015)	Chapter 22, 90
Dubai Customs	No	(Altaheri and Shaalan, 2020)	All HS-6:
DHL network	No	(Chen, Bromuri, <i>et al.</i> , 2021)	Chapter 70, 75, 80, 85, 90
African Customs	No	(Mai <i>et al.</i> , 2021)	Undefined
Korea Customs	No	(Lee <i>et al.</i> , 2021)	Chapter 85
Korea Customs Valuation and Classification Institute	No	(Lee <i>et al.</i> , 2024)	Chapter 84, 85, 90
Belgian Customs	No	(Vanhoeeyveld <i>et al.</i> , 2020)	-
Indonesian Customs Dataset	No	(Harsani <i>et al.</i> , 2020; Paramartha <i>et al.</i> , 2021)	Chapter 64
HUUB company	No	(Barbosa, 2021)	Chapter 61, 62, 63, 64, 42, 43
Chinese customs	No	(Zhou <i>et al.</i> , 2022)	Chapters relate to daily commodities, electronic products, chemical raw materials and fresh food
Brazilian Revenue Service	No	(de Lima <i>et al.</i> , 2022)	Chapter 90
e-Origin2	No	(Stassin <i>et al.</i> , 2023) (Amel <i>et al.</i> , 2023)	HS-6: 967 sub-heading HS-6: 16 sub-heading
Chinese commodity declaration	No	(Liao <i>et al.</i> , 2024)	HS-6: 26 sub-heading

Table 2.2 Summary of datasets utilised in previous studies on international trade.

⁷ <https://aws.amazon.com/marketplace/>⁸ <https://www.zauba.com/>

2.2 Classification Task in Trade Transactions

Product classification continues to present recurring challenges in selecting the correct code for trade products (Brigstock, 2017; Heijde, 2019; Lee *et al.*, 2024). A survey revealed that 91% of respondents experienced difficulties with product classification (KPMG and Reuters, 2016). Examples of terminological discrepancies between commercial product descriptions and HS terminology are presented in Table 2.3. Addressing this challenge typically requires domain-specific expertise because product classification is complicated by terminological gaps and the complex structure in a large number of product categories (Edirisinghe, 2017). This trade expertise is important to guarantee consistency and prevent ambiguous interpretations when assigning product codes (Grainger, 2024). Nonetheless, addressing this requirement entails extra financial cost to manage the resulting complexity (Heijde, 2019). On the other hand, the availability of automatic tools to assist users in selecting the appropriate HS code within the trade domain remains limited. Moreover, many existing systems exhibit functional shortcomings that constrain their practical applicability (Barbosa, 2021; Grainger, 2024; Heijde, 2019; Kappler, 2011).

Commercial Description	HS Description	Six-digit HS code
Laptop	Portable digital automatic data-processing machine	847130
Baby Food	Homogenised composite food preparation	190110
Wheelchair	Carriage for disabled persons, not mechanically propelled	871310
Salad spinner	Centrifuge	842119
Blow drier	Electro-thermic hairdressing apparatus	851632

Table 2.3 Examples of differences between technical HS code technical definitions and commercial terminology.

To establish an effective approach for developing a machine learning method for product classification, this section discusses relevant literature that has been studied for the classification task. In reviewing relevant studies, this section also provides an overview of their methods and mechanisms. The discussion examines the evolution of classification methods, encompassing classification algorithms, preprocessing techniques, and feature representation methods, ranging from traditional approaches

to modern language models. The main purpose of the steps mentioned before is to identify the gaps in the literature, which will be summarised at the end of this section.

2.2.1 Classification Methods

A classification task is a type of machine learning problem and part of a subcategory of supervised learning, where the goal is to predict a categorical class label of new data or a data point based on past observations (Aggarwal and Zhai, 2012; Raschka *et al.*, 2022). It is defined as text classification when the task is to categorise a group of documents into categorical class labels (Kadhim, 2019; Kowsari *et al.*, 2019). Formally, a classification task is defined as follows: given a learning dataset $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_c, y_c)\}$, where each instance x_i is labelled with one class y_i from a finite set of c discrete categories. When binary classification involves two categorical class labels, a multiclass classification task involves more than two class labels. The multiclass classification task is particularly applicable in our case for product classification in the trade domain, where each product description must be assigned to a single product category among thousands of HS codes.

Classification problems have attracted research attention in the domain of international trade over the past two decades. Early research focused on rule-based methods that relied on manually crafted rules and patterns for data processing. The earliest documented study was performed in 2004 (Singh and Sahu, 2004). This study develops a decision support system (DSS) based on several rules designed by the customs officers and employs fuzzy logic to recommend product classification. Another research study utilised an ontology-based approach for product category recommendation relying on if-then rules and pattern matching to select the most suitable category (Yang *et al.*, 2013). These early rule-based studies demonstrate clear advantages in terms of result interpretability and integration of knowledge with the domain expertise. However, the systems mentioned above tend to rely heavily on personal expertise in the system developing stage, which often resulting in divergent assessments and biased justifications. In addition, a key limitation of rule-based systems is their complexity when applied to complex scenarios, combined with limited adaptability and scalability (Cintra *et al.*, 2010).

Since machine learning generates superior accuracy and processing speed compared

to rule-based methods (Sarker, 2021), the adoption of this method has gradually increased in the field of international trade. Early machine learning research in the trade domain explored various approaches to improve model classification performance. Turhan found that combining a topic modelling technique based on Latent Dirichlet Allocation (LDA) with a visual matching module improved model performance using a limited number of product categories (Turhan *et al.*, 2015). Ding employed a background net, implemented as a weighted, undirected graph, to capture hierarchical relationships derived from word co-occurrence patterns in product descriptions. These relationships served as features for classifying product descriptions into product categories (Ding *et al.*, 2015). Comparative studies of machine learning approaches have yielded varying results in HS code classification tasks. Barbosa (Barbosa, 2021) found that Decision Trees (DT) and Random Forests (RF) had similar performance and outperformed Logistic Regression and Naïve Bayes performance using a private logistics company dataset, while Ryzhova (Ryzhova and Sochenkov, 2019) reported logistic regression's superiority over RF evaluated on Russian customs datasets. Altaheri applied Support Vector Machines (SVM) to capture non-linear decision boundaries, assessed on the Dubai customs dataset (Altaheri and Shaalan, 2020), where the model performance was superior to other machine learning algorithms such as DT, RF, AdaBoost, and Naïve Bayes classifiers. The discussed studies collectively have produced different empirical findings, suggesting that suitable algorithm selection and technical implementation contribute to achieving optimal model performance for trade product classification.

A transition has been made from traditional machine learning algorithms to a neural network approach, leading to notable enhancements (Li, Peng, *et al.*, 2022; Minaee *et al.*, 2021). Inputs in neural networks flow through hidden layers to capture complex, non-linear relationships, and as more layers are added, the network becomes deeper, which is coined the term deep learning (Goodfellow *et al.*, 2016; Hornik *et al.*, 1989; LeCun *et al.*, 2015). One of the fundamental neural network architectures, feedforward neural networks (FFNs), process inputs unidirectionally through hidden layers and are known as universal approximators that can theoretically model any continuous function. Combining this capability with innovative architectural designs, availability of abundant data, and hardware developments, deep neural networks have achieved many state-of-the-art results in various applications, including classification tasks

(Alzubaidi *et al.*, 2021; Bianco *et al.*, 2018).

A study of the product classification task in the international trade domain applied FFNs and proposed a method to predict HS codes given product declarations, which consist of more than 3,000 product categories (Ruder, 2020). The network comprised two sequential hidden layers, which were configured with 512-1024 nodes, and non-linear transformations to enable pattern recognition of text features at multiple abstraction levels through network optimisation using the RMSprop optimizer and hyperparameter adjustment. This study revealed that the FFN model achieved significant performance over traditional machine learning, which were utilised as baseline algorithms in the study.

Another type of neural network is Convolutional Neural Networks (CNNs), which excel in spatial hierarchy modelling to capture local patterns and utilise fewer parameters than fully connected networks (Alzubaidi *et al.*, 2021; LeCun *et al.*, 1989). CNNs use kernels to slide over the input to detect patterns and pooling layers to reduce the dimensionality of data before combining with fully connected layers for the final classification task (Goodfellow *et al.*, 2016).

Adaptation of CNNs in research studies on international trade has been utilised in different roles as feature extractors, end-to-end classifiers, or hybrid architecture components (Amel *et al.*, 2023; He *et al.*, 2021; Lee, Dongju *et al.*, 2020; Li and Li, 2019; Luppès, 2019; Ruder, 2020; Ryzhova and Sochenkov, 2019; Zhou *et al.*, 2022). As a hybrid component, CNNs were demonstrated in a study where the text features of product descriptions were extracted using CNNs and at the same time, it was utilised as a classifier to predict product categories. In this study, the kernel convolutional layers were configured with two sizes: 32 and 96 (Ryzhova and Sochenkov, 2019). CNNs have also been utilised to integrate multimodal data types composed of image and text data. A study by Li (Li and Li, 2019) introduced the implementation of a dual-CNN model trained on multimodal data combining product descriptions and images to predict the product category. Another study employed a ResNet-50 encoder to extract image features before concatenating them with text features from product descriptions to predict the HS code (Amel *et al.*, 2023). For extracting textual features, a CNN architecture was adopted in the text classification methodology in trade product classification, implementing convolutional layers with

filter sizes of 3, 4, and 5 (Yoon Kim, 2014). However, through hyperparameter optimisation configured in the network, it was revealed that models with filter sizes of 2, 3, 4, and 5 achieved the optimal classification accuracy (Luppens, 2019). Another study employed a customised CNN architecture to identify key product terms and reduce product description ambiguity for product classification (Zhou *et al.*, 2022). Finally, a previous study proposed the implementation of varied kernel and pooling configurations to extract text features, which stems from a language model to integrate into a dual classifier using a CNN framework (He *et al.*, 2021).

CNNs' adaptability has advanced the research in trade domain by facilitating efficient processing for both image and text data, nevertheless, they have limitations that influence their practical application. CNNs require manual configuration of the model architecture, which involve fine tuning of hyperparameters, where suboptimal design potentially leading to performance degradation (Zhang and Wallace, 2017). Additionally, CNN-based models have challenges in capturing long-range dependencies, resulting in the loss of temporal information (Mitkov, 2005) and inadequate modelling of semantic relations to short text sequences.

Unlike CNNs that specialised for spatial data types, Recurrent Neural Networks (RNNs) are a class of neural networks designed for sequential data, such as time series and text data (Rumelhart *et al.*, 1986). This network architecture process inputs sequentially, one step at a time, to preserve memory of previous information through hidden states updated via recurrent processing. Some advanced RNN variants, such as Bidirectional RNNs (Schuster and Paliwal, 1997), Gated Recurrent Unit (GRU) (Chung *et al.*, 2014), and Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), have been developed to address some limitations, such as gradient issues through gating mechanisms.

RNNs generally use a many-to-one architecture where a single output comes from input sequences to process in text classification applications. In this study, product trade classification applied this method successfully (Ryzhova and Sochenkov, 2019). Bi-LSTM architectures did outperform standard LSTM configurations in trade-related tasks, as was shown by implementing LSTM as well as Bidirectional LSTM (Bi-LSTM) models for product classification. Likewise, one study used a two-layer unidirectional LSTM for sequential features. The study integrated the output with spatial and textual

graph modules before final processing through fully connected layers for commodity category prediction (Du *et al.*, 2021). RNNs are strong in a fundamental way because they can hold internal states. They process sequential inputs by incorporating information from previous timesteps onward. However, their sequential nature causes computational issues and constrains their ability to find semantic links that are distant or non-adjacent in textual data (Li, 2024).

Sequence-to-Sequence (Seq2Seq) models are neural network architectures that are designed to transform input sequences into their corresponding output sequences, and they are widely implemented in machine translation systems (Sutskever *et al.*, 2014). The architecture consists of two fundamental components, an encoder and a decoder. The encoder processes input sequences and transforms them into internal representations, and the decoder subsequently utilises the representation to generate corresponding output sequences.

A study utilised Seq2Seq models to predict trade product categories, where the encoder processed product descriptions and the decoder predicted HS codes (Chen, Bromuri, *et al.*, 2021). The key mechanism of this method frames classification as a sequence-to-sequence problem, which translates product descriptions into HS codes. Seq2Seq models employ specialised components where the encoder focuses exclusively on understanding and compressing input sequences, while the decoder RNN specialises in sequence generation. However, traditional Seq2Seq models have limitations as the encoder's context vector often experiences a bottleneck effect, simultaneously retaining and conveying all necessary input information, which can result in suboptimal model performance.

2.2.2 Classification Approaches

Classification system design should align with the natural structure of the problem domain. This section examines the grounds for classification configuration and their implementation in previous studies in the international trade domain. In supervised learning contexts, classification tasks are generally categorised into flat and hierarchical classification (Jo, 2018). Flat classification provides a direct approach by assigning instances to class categories without considering relationships or hierarchical structures between classes (Gao *et al.*, 2020; Sun and Lim, 2001).

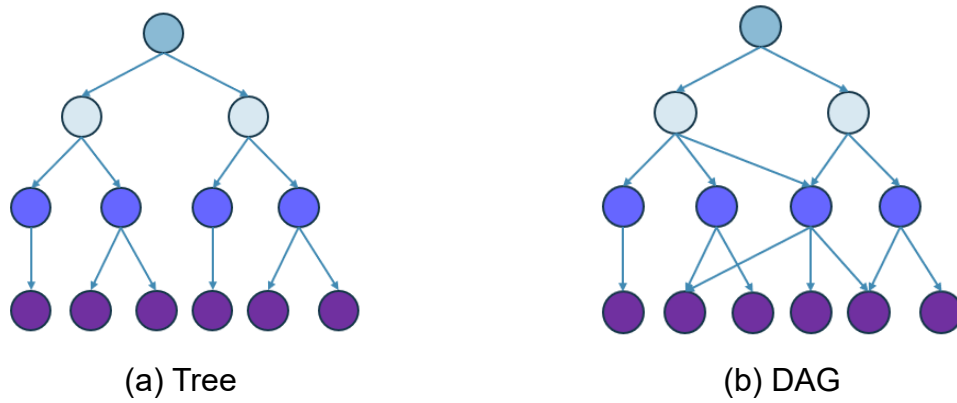


Figure 2.4 Comparison between a hierarchical structure of a Tree and a DAG.

According to Gao (Gao *et al.*, 2020), hierarchical classification can be broadly categorised into two primary approaches: local and global, each varying in how it leverages hierarchical structure during classification processes. A detailed explanation of each approach is provided below:

- a. Local binary classifier (Figure 2.5 a). This method handles each label as an independent class, completely disregarding hierarchical structure. It represents one of the simplest implementations and closely resembles the one-versus-rest classification strategy.
- b. Local per-level classifier (Figure 2.5 b). This method assigns separate classifiers to each hierarchical level. Each classifier functions in its respective layer, only making decisions at that specific level.
- c. Local per-parent classifier (Figure 2.5 c). This method assigns separate classifiers to each parent node, with each classifier responsible for selecting among its children. This approach facilitates the model to utilise partial hierarchical information during prediction processes.
- d. Global classifier (Figure 2.5 d). Unlike local methods, the global approach simultaneously uses the entire hierarchical structure to make a final classification decision.

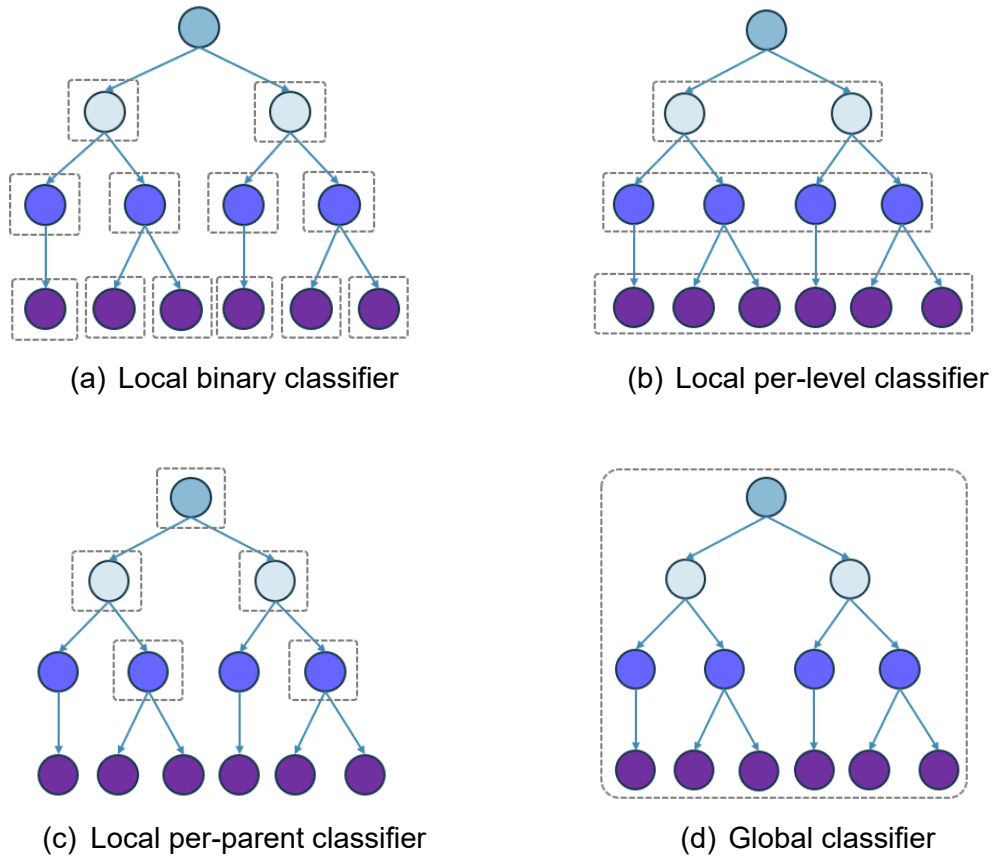


Figure 2.5 Types of classifiers used in hierarchical classification frameworks.

The implementation of trade product classification using hierarchical approaches has been demonstrated in a few studies. Chen implemented a hierarchical global approach where each node corresponds to an HS code (Chen, Bromuri, *et al.*, 2021). Barbosa (Barbosa, 2021) applied a local classifier per level approach, motivated by the low accuracy observed in flat classification and the potential advantages of using hierarchical classification.

The main difference between flat and hierarchical classification approaches resides in their treatment of class relationships and structural complexity. Structured classification systems, such as HS codes, contain embedded taxonomic information that reflects meaningful categorical relationships. The selection between flat and hierarchical classification approaches means a balance between simplicity and complexity, with hierarchical methods providing more complete predictions.

2.2.3 Text Preprocessing Techniques

Although text preprocessing plays an important role in NLP performance, it has often

been overlooked as deep learning research has shifted toward model architecture development and advanced training techniques (Camacho-Collados and Pilehvar, 2018; Naseem, Razzak and Eklund, 2021). This section outlines several standard text preprocessing techniques that have been utilised in studies in the trade domain and aims to provide an understanding of which of these techniques can be used in a classification task model to achieve optimal model performance.

Tokenisation has been used as part of text preprocessing besides the other techniques such as stop word removal, stemming or lemmatisation (Anandarajan *et al.*, 2018). This step is performed after lowercasing, which is implemented at the initial preprocessing phases, aiming to lessen input variance and often improves model performance (Uysal and Gunal, 2014). The traditional method usually utilises word-level tokens to split the text by spacing. However, modern language models utilise subword tokenisation strategies that segment the text into smaller units than word-level tokens, such as subwords or individual characters. This subword approach has enhanced vocabulary management to handle morphological word structures, unknown words, and provide better cross-language adaptability.

Product descriptions in trade declarations often contain of non-alphabetical characters. These characters include numbers such as [0–9], punctuation marks such as [.,!,?], and special characters such as [@, %, #, -]. Several characters in trade data usually do not signify functional information, unlike in other text classification domains such as sentiment analysis or hate speech detection, where punctuation often conveys emotional content or expresses sentiment (Siino *et al.*, 2021).

The contribution of numerical values within the text input is still questioned, particularly regarding its contribution for semantic understanding and model performance. Evidence supports the argument to preserve numerical data in order to improve model performance (Denny and Spirling, 2018), whilst contrasting research indicates that numbers are better removed during preprocessing steps (Anandarajan *et al.*, 2018). Removing attributes or parts of data from domain-specific fields like international trade may lead to information loss. Numerical values can often convey important context-specific meaning in an intuitive way.

Noise, as well as stop words, frequently occur within textual data, obscuring key

information components. Stop words such as ‘and’, ‘the’, ‘be’, ‘to’, and ‘of’ provide minimal information value, since they are easily identifiable in well-structured grammatical sentences (Wilbur and Sirotkin, 1992). However, in product descriptions, stop words may function as supplementary information related to core product characteristics, including units of measurement (weight or length) or material specifications. It is considered practical to remove digits, punctuation, and other non-essential elements from product descriptions, as such components may introduce noise and compromise the clarity of primary textual content. However, it remains an open research question whether such information existed in product descriptions that contain important contextual signals, and whether these signals contribute to accurate text analysis.

Various preprocessing techniques implemented across multiple product classification studies are presented in Table 2.4, which primarily involve text transformation and raw text data removal procedures. While several deep learning-based studies (Amel *et al.*, 2023; Chen, Bromuri, *et al.*, 2021; Li and Li, 2019; Pain, 2021; Ruder, 2020; Stassin *et al.*, 2023) opt to remove numerical digits during preprocessing phases, alternative studies (Marra de Artiñano *et al.*, 2023; Ruder, 2020) choose to retain numerical information by converting digits into their corresponding word representations. Stemming, which removes word endings to derive the root form, and lemmatization, which reduces words to their dictionary base form based on grammatical context, have been applied in previous studies on product declarations (Altaheri and Shaalan, 2020; Harsani *et al.*, 2020; Heijde, 2019). However, the impact of these techniques appears to be relatively small, likely because the text already contains the base form of the product name. Additionally, some studies did not explicitly report the specific text preprocessing procedures (He *et al.*, 2021; Liao *et al.*, 2024).

Text processing	Study
Converting to lowercase	(Marra de Artiñano <i>et al.</i> , 2023) (de Lima <i>et al.</i> , 2022) (Chen, Bromuri, <i>et al.</i> , 2021) (Harsani <i>et al.</i> , 2020) (Ruder, 2020) (Luppès, 2019)
Stemming	(Harsani <i>et al.</i> , 2020) (Heijde, 2019)
Lemmatisation	(Shubham <i>et al.</i> , 2023) (Marra de Artiñano <i>et al.</i> , 2023) (Paramartha <i>et al.</i> , 2021) (Ruder, 2020) (Altaheri and Shaalan, 2020) (Heijde, 2019)
Removing duplicated words	(Paramartha <i>et al.</i> , 2021) (Spichakova and Haav, 2020a) (Altaheri and Shaalan, 2020) (Ding <i>et al.</i> , 2015)
Removing abbreviation	(Shubham <i>et al.</i> , 2023)
Removing punctuation	(Stassin <i>et al.</i> , 2023) (Amel <i>et al.</i> , 2023) (Paramartha <i>et al.</i> , 2021) (Chen, Bromuri, <i>et al.</i> , 2021) (Ruder, 2020) (Altaheri and Shaalan, 2020) (Li and Li, 2019) (Heijde, 2019) (Ding <i>et al.</i> , 2015)
Removing special characters	(Stassin <i>et al.</i> , 2023) (Shubham <i>et al.</i> , 2023) (Amel <i>et al.</i> , 2023) (Luppès, 2019)
Removing non-language dataset (ex., English, Spanish)	(Marra de Artiñano <i>et al.</i> , 2023) (Paramartha <i>et al.</i> , 2021) (Ruder, 2020) (Altaheri and Shaalan, 2020)
Removing stop words (on, the, in, of)	(Marra de Artiñano <i>et al.</i> , 2023) (Paramartha <i>et al.</i> , 2021) (Pain, 2021) (Spichakova and Haav, 2020a) (Harsani <i>et al.</i> , 2020) (Ruder, 2020) (Altaheri and Shaalan, 2020) (Li and Li, 2019) (Heijde, 2019) (Ding <i>et al.</i> , 2015) (Heijde, 2019)
Removing noise (box, code, gram)	(Marra de Artiñano <i>et al.</i> , 2023) (Pain, 2021) (Ruder, 2020) (Li and Li, 2019)
Removing numbers/digits	(Stassin <i>et al.</i> , 2023) (Paramartha <i>et al.</i> , 2021) (Amel <i>et al.</i> , 2023) (Pain, 2021) (Chen, Bromuri, <i>et al.</i> , 2021) (Spichakova and Haav, 2020a) (Ruder, 2020) (Altaheri and Shaalan, 2020) (Li and Li, 2019) (Ding <i>et al.</i> , 2015)
Converting numbers to text	(Marra de Artiñano <i>et al.</i> , 2023) (Ruder, 2020)
Removing company information	(Shubham <i>et al.</i> , 2023) (de Lima <i>et al.</i> , 2022)

Table 2.4 Preprocessing techniques implemented in trade transactions.

2.2.4 Text Representation Methods

The transformation of raw text into a structured numerical representation is a

fundamental prerequisite for text classification tasks (Kadhim, 2019; Kowsari *et al.*, 2019; Li, Peng, *et al.*, 2022). The quality of text representation in capturing relevant features determines the effectiveness of the method (Bengio *et al.*, 2013; Szymański, 2014). Over the past decade, numerous scholarly efforts have developed methods for vectorising text input, with the primary objective of generating meaningful numerical representations of text (Goldberg, 2017).

This section reviews the developments of text representation consisting of traditional methods and modern language models, which continued by reviewing their application in several studies in international trade.

Sparse Representation

One-hot encoding (OHE) is a straightforward method for input encoding. When working with text data, this method changes the words into vectors of zeros and ones, where each vector has only one '1' and all other positions are '0', and the vector length equals the number of words in the vocabulary (Goldberg, 2017). In the study of product classification, Barbosa (Barbosa, 2021) utilised OHE to encode elements in product declarations, drawing on a dataset specific to fashion that incorporated features such as gender, material, age suitability, and fabric type. OHE is recognised as an efficient computation method. However, this method has limitations when dealing with a large vocabulary, characterised by increasing vector dimensionality that scales linearly with unique token count, thus decreasing computational feasibility. This method has limitations in capturing semantic relationships between words, which reduces the ability to represent complex linguistic structures.

Unlike OHE, Bag-of-Words (BoW) is a count-based vector representation that incorporates word frequencies, creating a histogram-like representation of words within text (Goldberg, 2017; Zhang *et al.*, 2010). The BoW has an extended feature that functions to group n-tokens in word sequences, which is referred to as bag-of-n-grams, where n denotes the token count. Harsani (Harsani *et al.*, 2020) utilised BoW with bigram features to represent product descriptions for product classification. This strategy enabled BoW to represent and recognise phrases within product descriptions, instead of treating the words independently, and the findings presented that bigrams outperformed unigrams in classification performance using trade data.

BoW offers some advantages in capturing the information of word frequency from texts, which can signal the theme or topic of the text data and still offers efficient computation. However, it has similar limitations to OHE, in that it struggles with sparsity issues and limited ability to model semantic relationships between words. Although n-gram approaches (particularly bigrams) often provide superior representation compared to standard BoW (Goldberg, 2017), larger n-grams introduce additional computational complexity and out-of-vocabulary challenges.

Term Frequency-Inverse Document Frequency (TF-IDF) is a representation method that adjusts word weights based on their importance within individual documents and across document collections (Robertson, 2004; Sparck Jones, 1972). It emphasises unique or distinctive words while reducing the significance of common words that appear frequently across documents. TF-IDF has been implemented in trade-related studies (Altaheri and Shaalan, 2020; Paramartha *et al.*, 2021; Ryzhova and Sochenkov, 2019) because it effectively reduces the influence of high-frequency and low-informative terms, and intensifying the impact of more discriminative words in using text data, in this case, product descriptions.

TF-IDF is effective in highlighting informative words, but its limitation lies in the inability to account for semantic relationships between words. However, this characteristic makes TF-IDF more suitable than BoW or OHE for domains that rely heavily on specialised terminology, where term distinctiveness is more valuable in this context.

Dense Representations

To address common issues associated with sparse text representations, which mainly consist of zeros. An innovative method was introduced to represent words as distributional representations in continuous vector spaces (Bengio *et al.*, 2003). Dense representations, commonly referred to as embeddings, reduce the dimensionality of features. While dense representations utilise hundreds of dimensions of real-valued vectors, sparse representations use thousands of dimensions to implement a distributional hypothesis that words in similar contexts have similar meanings (Harris, 1954). The Neural Probabilistic Language Model (NPLM) was initialised to generate dense word embeddings using neural networks (Bengio *et al.*, 2003), to represent semantic relationships in embedding spaces, offering advantages by improving

generalisation capabilities and more efficient computation due to their lower dimensionality (Goldberg, 2017).

Non-contextual Word Embedding

Word2Vec (Mikolov *et al.*, 2013) is a well-known method for generating word embeddings. The architecture is composed of a shallow three-layer network and has two alternatives to train models: Continuous Bag-of-Words (CBOW) and Skip-gram. CBOW has a mechanism to predict a target word from its surrounding context, while the Skip-gram predicts context words from a single input word.

In the trade domain, the application of Word2Vec in product classification was utilised to extract text features from product descriptions before feeding the input into several classifiers (Ruder, 2020). Training language models on custom datasets revealed a finding that Skip-gram produced slightly better results than CBOW. On the other hand, Luppès (Luppès, 2019) used a pre-trained Word2Vec model trained on Google News to produce 300-dimensional embeddings and reported strong results on a comparable dataset.

GloVe (Global Vectors) is another word embedding method, and the algorithm is based on a study of Pennington (Pennington *et al.*, 2014). It is a count-based word embedding method that generates word embeddings by leveraging a co-occurrence matrix to capture global statistical information about word usage. The extended study performed by Ruder (Ruder, 2020) utilised GloVe to extract text features from product descriptions, revealing that both Word2Vec and GloVe produce comparable model performance using a similar classifier. In this study, the pretrained models⁹ of GloVe were trained by the product description since it delivered better performance than the off-the-shelf version.

FastText (Bojanowski *et al.*, 2017) introduces a key innovation by treating each word as a collection of n subword components to generate embeddings, while the learning method is similar to that of Word2vec with the Skip-gram. FastText method processes the word 'laptop' to have tokens as ['<la', 'lap', 'apt', 'pto', 'top', 'op>'] with trigrams of subwords and the representation is the sum of the subwords' vectors. FastText also

⁹ <https://nlp.stanford.edu/projects/glove/>

provides pre-trained models in different sizes that can be obtained from its public repository¹⁰.

Word2Vec and GloVe perform well at capturing word-level semantics, but they have limitations when dealing with morphological variations, rare words and compound words, which are often absent from training vocabularies. The out-of-vocabulary (OOV) problem is typically addressed by replacing unknown words with generic tokens (<UNK>), resulting in information loss. FastText addresses this limitation by utilising subword representations, presenting improved generalisation for unseen terms. However, it needs higher computation and possibilities to produce similar embeddings for unrelated words with similar morphological structures.

Contextual Word Embedding

Non-contextualised word embedding methods, such as Word2Vec, GloVe, and FastText, assign the same fixed embedding to a word, despite having different contexts, which makes it challenging to represent words with multiple meanings or polysemy. Peters (Peters *et al.*, 2018) developed Embeddings from Language Models (ELMo), a bidirectional LSTM model that produces context-sensitive embeddings to overcome this limitation. Unlike static word embeddings, ELMo produces word embeddings that change according to how the word is used in a sentence or based on the context, representing a significant step forward in language modelling. ELMo facilitates the embeddings from a pretrained model being used for downstream tasks for a specific task (Chelba *et al.*, 2014). In a study focused on product classification using Russian product descriptions, ELMo word embeddings from the DeepPavlov library were employed to create textual representations (Ryzhova and Sochenkov, 2019). Unexpectedly, the model performance using ELMo was lower than that of Word2Vec, though this study used different classifiers, which may not be comparable.

Transformer Architecture

The Transformer architecture has revolutionised sequence modelling (Vaswani *et al.*, 2017), as it eliminates recurrence and instead relies entirely on self-attention, which in turn allows input sequences to be processed in parallel, making them suitable for

¹⁰ <https://fasttext.cc/docs/en/crawl-vectors.html>

modern hardware acceleration. With the Transformer, the model can dynamically focus on the most informative part of the input using a self-attention mechanism built on earlier attention mechanisms (Bahdanau *et al.*, 2016; Luong *et al.*, 2015). Transformer architecture captures long-range dependencies through key architectural components, such as multi-head attention and positional encoding, and their parallel processing capabilities. A wide range of models has been addressed through the flexible configurations of the Transformer architecture (Devlin *et al.*, 2019; Dosovitskiy *et al.*, 2021; Lin *et al.*, 2021). Transformer-based models can be implemented using several options, including encoder-only architectures, decoder-only architectures, and encoder–decoder architectures. For language models, this flexibility enables the capture of subtle and complex syntactic and semantic patterns across applications. To implement a Transformer-based architecture to build a model for a specific task, it is typically driven by model-specific goals in the learning phase and architectural design with scale and domain of training data. The following section outlines the implementation of Transformer-based architecture in several language models, divided mainly into encoder-based and decoder-based models.

Transformer Encoder-Only Models

Bidirectional Encoder Representations from Transformers (BERT) is a widely known and most influential transformer-based model (Devlin *et al.*, 2019). It was constructed based on the encoder component of the Transformer architecture and performs well on tasks that require deep contextual interpretation of text and natural language understanding (NLU), such as text classification. BERT was trained using dual pretraining objectives, where the first objective is to mask tokens in input sequences randomly and train the model to predict them based on the surrounding context, termed as masked language modelling (MLM). The second objective is next sentence prediction (NSP), which trains the model to determine whether two consecutive sentences follow each other. BERT has been used in a study of trade product classification, where BERTimbau was utilised to extract text features of product description to predict HS code (de Lima *et al.*, 2022). BERTimbau is a BERT variant that was specifically designed for the Portuguese language (Souza *et al.*, 2020).

After the BERT model was introduced, numerous variants inspired by BERT have been developed and made publicly available. Below are several language models that

will be used in the previous chapter of this thesis.

To improve model efficiency, A Lite BERT (ALBERT) (Lan *et al.*, 2020) employs cross-layer parameter sharing and factorised embeddings and replaces BERT's NSP with a Sentence Order Prediction (SOP) task. This approach results in a substantial reduction in the number of parameters of the model, where the ALBERT-base model has 12 million parameters compared to 110 million in BERT-base.

DistilBERT was reported to be 40% smaller and 60% faster at inference time, while achieving approximately 97% of BERT's performance (Sanh *et al.*, 2020). The key innovations behind DistilBERT were driven by knowledge distillation (Hinton *et al.*, 2015) to guide model training, reducing the number of encoder layers by half, removing the NSP task, and employing customised loss functions to optimise model parameters.

DeBERTa (Decoding-enhanced BERT with Disentangled Attention) is a BERT variant that has key innovations, namely the Disentangled Attention Mechanism (He *et al.*, 2023). It works by disentangling the representation of each word into content and relative positional vectors, rather than using absolute positional encodings, which improves the model's ability to understand relational dynamics between words based on their relative positions.

ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) proposes Replaced Token Detection (RTD) as a substitute for BERT's MLM (Clark *et al.*, 2020). It uses a generator-discriminator framework to replace some tokens with possible alternatives and trains the model to identify which tokens are original versus replaced across the entire sequence to learn from all token positions instead of a specific percentage that are masked like BERT.

Robustly Optimized BERT Pretraining Approach (RoBERTa) (Liu *et al.*, 2019) is an optimised variant of BERT, developed under the assumption that the original BERT model was undertrained. RoBERTa employs a different tokenisation algorithm and introduces dynamic masking, in which the masking patterns are generated on the fly during pre-training. In addition, RoBERTa improves training stability and model performance by using larger batch sizes and pre-training on a more diverse and extensive corpus.

Transformer-based language models, especially those employing the encoder architecture, are recognised for their ability to generate semantically rich textual representations, making them valuable for improving the accuracy of downstream classification tasks. Nevertheless, with a wide range of pretrained models available, it is necessary to identify the most suitable baseline for feature extraction in domain-specific contexts, such as trade classification. Examining the most optimal language models is necessary to represent trade data, considering that these transformer-based models have been pre-trained on extensive and general-purpose corpora utilising different model architectures and objectives, which differ from the characteristics of domain-specific trade datasets, as summarised in Table 2.5.

Transformer Decoder-Only Models

One of the most prominent implementations of the Transformer architecture for the decoder module is Generative Pre-Training, also known as the Generative Pretrained Transformer (GPT) (Radford *et al.*, 2018), which is primarily designed for natural language generation (NLG) tasks. This language model highlighted for its ability to generate human-like text sequence through next-word prediction or next-token prediction (NTP). Generative models are well-suited for text generation since they tend to be less effective for downstream classification tasks compared to transformer-based models that use an encoder-only component, such as BERT (Li, Li, *et al.*, 2023). However, generative models have recently gained increased attention due to their ability to perform few-shot and zero-shot learning through prompt-based responses.

ChatGPT is one of the examples of a generative model built on a transformer decoder component. One of its versions, ChatGPT 3.0, has been used in international trade to predict HS codes based on queries of product descriptions. However, the findings revealed that this method was not suitable for the product classification task due to frequent inaccuracies of the outputs (Hussain and Ahmed, 2023). Building on a similar study, a newer version of the model, ChatGPT 3.5, was utilised for the same task and revealed that the model underperformed when compared to a supervised classification method (Marra de Artiñano *et al.*, 2023).

Despite generative models are not directly comparable to NLU models in terms of classification accuracy, they provide additional benefits, such as providing

explainability, transparency in decision-making and offering more flexible and human-like interactions. For instance, Navasardyan (Navasardyan, 2024) demonstrated the application of generative models to HS code assignment with integrated natural language explanations. A key advantage of such models lies in their capacity for flexible zero-shot and few-shot learning, alongside their ability to generate interpretable classification rationales without task-specific fine-tuning. Nonetheless, the scalability of this approach is constrained by the substantial service costs associated with processing high volumes of trade transactions, and privacy concerns arise when sensitive trade documentation is handled via API-based deployments.

LMs	Study	Architecture	Pre-Training Task	Corpus Used
Word2vec	(Mikolov <i>et al.</i> , 2013)	Skip-Gram, CBOW prediction	Skip-Gram/CBOW prediction	Google News
GloVe	(Pennington <i>et al.</i> , 2014)	Matrix Factorization	Log-bilinear co-occurrence	Common Crawl (CC) corpus
FastText	(Bojanowski <i>et al.</i> , 2017)	Skip-Gram, CBOW	Character n-gram prediction	Multiple corpora
ELMo	(Peters <i>et al.</i> , 2018)	Bi-LSTM	Bi-LM	1 billion Word Benchmark
BERT	(Devlin <i>et al.</i> , 2019)	Transformer Encoder	MLM + NSP	Wikipedia (English) + Book-Corpus
RoBERTa	(Liu <i>et al.</i> , 2019)	Transformer Encoder	MLM	BookCorpus + Wikipedia + CC-News + OpenWebText + Stories
ALBERT	(Lan <i>et al.</i> , 2020)	Transformer Encoder	MLM + SOP	Wikipedia (English) + Book-Corpus
XLNet	(Yang <i>et al.</i> , 2019)	Auto-regressive Transformer Encoder	Permuted Language Modelling	Wikipedia (English) + Book-Corpus + Giga 5 + Clue-Web + CC
ELECTRA	(Clark <i>et al.</i> , 2020)	Transformer Encoder	RTD + MLM	Wikipedia (English) + Book-Corpus + Giga 5 + Clue-Web + CC
DistilBERT	(Sanh <i>et al.</i> , 2020)	Transformer Encoder	MLM + NSP	Wikipedia (English) + Book-Corpus
DeBERTa 3	(He <i>et al.</i> , 2023)	Transformer Encoder	MLM + enhanced attention	Wikipedia (English) + Book-Corpus + CC-News + OpenWebText
GPT-3.5	(Brown <i>et al.</i> , 2020)	Transformer Decoder	NTP	Common Crawl, OpenWebText2, Book-corpus, and Wikipedia

Table 2.5 A comparison of language models based on Transformer architecture.

LMs	Syntactical	Semantics	Context	Out of Vocabulary
One-Hot encoding	-	-	-	-
BoW	-	-	-	-
TF-IDF	-	-	-	-
Word2vec	✓	✓	-	-
GloVe	✓	✓	-	-
FastText	✓	✓	-	✓
ELMo	✓	✓	✓	✓
BERT	✓	✓	✓	✓
RoBERTa	✓	✓	✓	✓
ALBERT	✓	✓	✓	✓
XLNet	✓	✓	✓	✓
ELECTRA	✓	✓	✓	✓
DistilBERT	✓	✓	✓	✓
DeBERTa	✓	✓	✓	✓
GPT-3.5	✓	✓	✓	✓

Table 2.6 Key characteristics of sparse versus (non-)contextualised text representations.

2.2.5 Trade Product Classification with Word Embeddings

A comprehensive literature review has been performed related to product classification in the trade domain. The review encompasses developments from traditional methods to existing studies on deep learning-based text representation, summarised in Table 2.6. It identifies the key characteristics of text encoders along with their features. The review further examines a range of approaches that have been used in downstream text classification, including preprocessing techniques and classification methods, which are framed for international trade product classification.

Overall, evidence from the literature highlights key challenges in systematically conducting product classification, beginning with determining the optimal way to represent trade data and developing a classification method to enhance model performance. Although extensive studies on product classification mostly involve preprocessing methods, there has been limited focus on systematic approaches for conducting preprocessing techniques specifically for trade product data. Product description often contains diverse product attributes and multiple units of measurement, presented in inconsistent text formats, which require preprocessing

before further processing. However, any removal or modification of elements in the raw text must be carefully assessed for its potential impact. The choice to select preprocessing techniques requires systematic evaluation, because these decisions can substantially affect model performance, even when transformer-based models are employed to represent text input (Siino *et al.*, 2024).

Another gap in trade classification applications involves the limited use of contextual word embeddings, particularly those produced by transformer-based architectures. Despite empirical evidence demonstrating that such methodological approaches possess the potential to outperform traditional classification approaches and achieve strong performance outcomes in downstream classification task implementations (Fields *et al.*, 2024; Garrido-Merchan *et al.*, 2023; Subakti *et al.*, 2022), the use of contextual word embedding methodologies remains underutilised within the international trade domain context. With the increasing availability of diverse pretrained language models, an important question concerns which architecture is most appropriate for product description, given the specific linguistic and structural characteristics of such trade product data. Finally, selecting the most suitable pretrained model is essential to maximise predictive performance.

Prior studies generally apply a single classification approach, typically flat classification. It is necessary to understand whether flat or hierarchical approaches are optimal for trade product classification. To address this gap, this study aims to conduct a comparative evaluation to determine the optimal approach, following several studies that reported conflicting results regarding the effectiveness of both approaches (Cotacallapa *et al.*, 2024; Rezende *et al.*, 2022).

This thesis aims to address the problem of classifying product categories by developing a text classification pipeline composed of three main areas. First, the limited preprocessing strategies employed in studies in the trade domain require evaluation to determine which preprocessing steps have the most significant impact on model performance. Second, modern language models, such as transformer-based models, are promising for improving model performance. However, selecting the optimal model is necessary for adaptation with trade data to determine the optimal model. Finally, previous studies have mostly employed flat classification, whereas hierarchical classification is necessary to develop and evaluate their effectiveness in

comparison to the flat classification approach.

2.3 Sentence Embedding in Trade Transactions

In the previous section, we discussed word embeddings that represent individual words, while in this section we change our focus to sentence embeddings that aim to encode complete sequence information in continuous numerical spaces. Some methods commonly use aggregation of word embeddings or transformation methods to capture entire text sequences (Mishra and Viradiya, 2019; Ramesh Kashyap *et al.*, 2024). Word vector averaging is considered a common operation that is used to generate sentence-level embeddings (Wieting *et al.*, 2016). However, this operation causes word order information loss, which is important to preserve for maintaining syntactic and semantic sequence dependencies (Wieting *et al.*, 2016).

The skip-thought model was dedicated to generate sentence-level embeddings using a mechanism inspired by the skip-gram method in Word2Vec (Kiros *et al.*, 2015). The model utilised an RNN encoder-decoder architecture to process sentences sequentially, preserving sentence structure and order. Another sentence-level embedding model is Doc2Vec, an extended version of Word2Vec (Le and Mikolov, 2014). The model processes documents to capture document themes by learning word-level relationships in documents while simultaneously learning document vectors that represent the context in which those words appear. Two main approaches of Doc2Vec are Paragraph Vector Distributed Memory (PV-DM), which predicts words using the document vector and Distributed Bag-of-Words (PV-DBOW), which incorporates both document and word vectors for learning sentence embeddings.

Doc2Vec was utilised to extract sentence-level embeddings from product descriptions in a study on HS code classification (Spichakova and Haav, 2020a). This study implemented the PV-DBOW architecture to train the model, where several hyperparameters were required to be configured, including window size and embedding dimension, to improve the quality of the sentence representations. Additionally, training a Doc2Vec model from scratch requires a sufficiently large number of trade documents to produce meaningful embeddings before feeding them to a classifier.

Transformer-based models, such as BERT generate sentence embeddings

incorporating a special [CLS] token that is obtained from the aggregation of sentence-level embeddings during the model pretraining phase. However, this approach was criticised for producing embeddings that were suboptimal for the semantic similarity task (Ha *et al.*, 2021; Wang and Kuo, 2020). SBERT (Reimers and Gurevych, 2019) overcomes this limitation by presenting a framework where a pretrained transformer model is employed as the backbone encoder. Through its architecture that employs Siamese networks (bi-encoder), it substantially reduces computational overhead for sentence similarity tasks compared with the cross-encoder approach.

Numerous pretrained SBERT models are publicly available¹¹, and these off-the-shelf models can be readily used to generate sentence embeddings, as demonstrated in a study of HS code classification (Pain, 2021). This study used pretrained SBERT, which had backbone encoders derived from BERT and XLM-RoBERTa model variants, to generate sentence embeddings from product descriptions. These generated embeddings were then compared using sentence text similarity (STS) against a lookup table containing reference product descriptions from the HS guidelines, which were vectorised with the same model. The top-k most similar matches defined the corresponding HS codes. In this study, SBERT was used as a feature extractor without fine-tuning on trade-specific data. However, SBERT implementation, which were applied directly from public repository without further fine-tuning on domain-specific datasets are expected to produce representations that are suboptimal for downstream downstream classification tasks.

The Universal Sentence Encoder (USE) (Cer *et al.*, 2018) is a method for generating sentence-level embeddings, where the architecture is divided into two different categories. The first model version was based on the Deep Averaging Network (DAN), which is optimised for computational efficiency and well-suited to applications requiring fast inference and a Transformer-based variant, which is more resource-intensive but delivers superior performance on semantically complex tasks. Both models are available as pre-trained implementations in the TensorFlow repository¹².

USE was applied to extract text features from product declarations into sentence

¹¹ <https://huggingface.co/sentence-transformers>

¹² <https://www.tensorflow.org/hub>

representations utilising an unsupervised learning framework to evaluate the accuracy of assigned HS codes (Chen, Ben van Rijnsoever, *et al.*, 2021). In this method, sentence embeddings in each product category were aggregated into centroid vectors, which were stored in a lookup table. During inference, unseen product descriptions were compared to the stored centroid vectors, and the HS code prediction was selected from the one with the highest cosine similarity values.

2.3.1 Trade Product Classification with Sentence Embeddings

In comparison to word embeddings, sentence embeddings offer the advantage of representing full text sequences and retaining their semantic context. With this characteristic, sentence embeddings offer flexibility to encapsulate the embedding information into a single sentence, regardless of the sequence length. A single representative of embeddings for a sentence offers advantages for the sentence text similarity comparison and semantic similarity search (Reimers and Gurevych, 2019). It also enables machines to understand high-level understanding of granularity for text clustering, which in turn improves downstream task performance, such as for text classification (Conneau and Kiela, 2018) and sentiment analysis (Xu, Xie, *et al.*, 2023).

Generating sentence-level embeddings through word-embeddings aggregation or pooling layers offers both computational efficiency and baseline effectiveness (Wieting *et al.*, 2016). However, the detrimental effect of this conventional method by aggregating word representations becomes apparent when dealing with sentences that have similar word or character compositions but convey different meanings, which is prevalent in the trade product description (Luppess, 2019; Ruder, 2020; Ryzhova and Sochenkov, 2019). Doc2Vec aims to handle the issue by utilising sentence-level representation as it has been applied in a trade product classification study (Spichakova and Haav, 2020a). However, Doc2Vec lacks generalisation capability, as it typically requires training from scratch on the target corpus to produce effective representations (Lau and Baldwin, 2016). Additionally, Doc2Vec exhibits limitations in capturing semantic meaning when compared to more advanced language models. The emergence of sophisticated sentence embedding models like SBERT and USE addresses these fundamental shortcomings by learning to map sentences into a semantic space. The implementation of such models was conducted also by Pain (Pain, 2021) and Cer (Cer *et al.*, 2018) for trade product classification tasks. However,

both studies solely relied on pretrained models without fine-tuning on domain-specific datasets. In this case, the models are likely to produce suboptimal text representations, and the embeddings potentially suffer from anisotropic issues.

SBERT and USE with transformer-based models as backbone encoders show anisotropic characteristics when they are implemented without domain-specific fine-tuning procedures (Rajaei and Pilehvar, 2021). However, the manifestation of this phenomenon is less pronounced in comparison to vanilla BERT implementations (Li, Zhou, *et al.*, 2020). Anisotropy refers to non-uniform embeddings in which the vectors cluster within a narrow region rather than being evenly distributed in the embedding space. By contrast, isotropy is typically regarded as being beneficial for large language models since it enhances the representations and in turn improves downstream task performance (Rudman and Eickhoff, 2024). This anisotropic effect is often most evident in the upper layers of transformer-based architectures, where contextualisation processes significantly shape the distribution of embeddings (Liang *et al.*, 2021). Such clustering can cause semantically unrelated vectors to appear overly similar, reducing representational quality and complicating the separation of classes in downstream classification tasks. While most findings associate anisotropy with performance degradation, conflicting evidence exists from studies reporting the opposite, showing that increasing anisotropy, even at the cost of reduced isotropy, can yield performance gains across models and tasks (Rudman and Eickhoff, 2024). This leaves a gap in understanding how anisotropy manifests when implementing transformer-based models using our dataset and methodology.

Improving the quality of domain-specific classification tasks through the implementation of sentence embeddings has potential for improving downstream classification tasks. Despite well-established fine-tuning transformer-base model show a notable performance gains, generating sentence embeddings offers several advantages, ranging from the flexibility to generate a fixed-length embeddings, which offers flexibility. The emergence of sentence embedding models like SBERT and USE provides advantages by learning to map sentences into a semantic space where similar meanings are positioned closer together, regardless of the individual words used or their surface-level arrangement. However, in the absence of a fine-tuning approach for the utilisation of sentence embeddings derived from a pretrained

transformer-based model performed in these studies (Chen, Ben van Rijnsoever, *et al.*, 2021; Pain, 2021), it is expected that optimal performance is challenging to achieve. This issue can negatively affect performance in downstream applications and addressing it represents a key objective of this thesis.

Utilisation of sentence embedding as a representation of declared product description, potentially to enable capturing the semantics of the entire representation of the declared product information. Therefore, one of the promising options is a fine-tuning approach with a domain dataset to enhance the sentence embeddings by performing a contrastive learning paradigm (Xu, Xie, *et al.*, 2023). Contrastive learning has a concept to perform a transformation of the representation by pulling the similar sentences into the same categories and pushing away those that have different categories. In this study, our goal is to enhance sentence embeddings by fine-tuning them with the implementation of a contrastive learning paradigm. With this purpose, the result is expected to improve the embedding representation by alleviating the anisotropy issue and improving the classification performance.

2.4 Anomaly Detection in Trade Transactions

Having discussed text classification for trade product declaration, in this section, we shift our focus to detecting anomalies within trade transactions.

Anomalies, often known as outliers, abnormalities, rare events, or deviants, are terms commonly used within the academic literature (Chandola *et al.*, 2009). According to Hawkins (Hawkins, 1980), an outlier refers to a data point that significantly differs from other observations and may potentially originate from a different underlying data-generation process. The process of identifying an anomalous pattern that does not conform to the expected behaviour is referred to as anomaly detection (Chandola *et al.*, 2009). Early implementations of anomaly detection algorithms focused on traditional data mining applications (Agrawal and Agrawal, 2015). However, anomaly detection has been utilised in the modern deep learning-based approach that is capable of handling complex, high-dimensional, and heterogeneous data (Chalapathy and Chawla, 2019; Pang *et al.*, 2022).

In international trade, anomalous transactions may indicate fraudulent activities, which are defined as any attempt to deceive and evade, wholly or partly, the payment of

taxes or duties to obtain an advantage contrary to customs law or referring to customs evasion (World Customs Organization, 2018c). It contains several categories (Working Group on Commercial Fraud, 2010), such as under- or overvaluation, misclassification, smuggling, and falsification of origin. Misclassification refers to the submission of incorrect identification of a product's code (HS code) in customs declarations during the clearance process (Roman *et al.*, 2006). Several factors contribute to misclassification and pose a significant challenge in customs administration (Lushi and Gashi, 2016). Misclassified product categories can be caused by several factors, such as honest mistakes by the users, like manual errors (typos) because of a lack of knowledge, the result of improper assisted tools, or driven by an intention to obtain unlawful benefits (Heijde, 2019; Lombardi *et al.*, 2013) (Kappler, 2011). For the latter reason, the initiation of misclassification is often motivated by the discrepancies in tariff rates across different products (Betz, 2019; Che *et al.*, 2025; Mengistu *et al.*, 2019). The products that fall under lower tax rate categories are increasingly appealing to fraudsters looking to misclassify their products (Javorcik and Narciso, 2017). As a consequence, the authorities interpret such errors as instances of noncompliance, which may result in financial penalties and lead to delays in customs clearance, or even criminal charges (Heijde, 2019; Lombardi *et al.*, 2013).

Product misclassification presents significant risks to customs mechanisms because it can lead to the underpayment of duties and taxes (Global Financial Integrity, 2021; Lushi and Gashi, 2016). According to the Canada Auditor's report (Office of The Auditor General of Canada, 2017), 20% of products entering Canada during the 2015–2016 fiscal year were incorrectly classified. Similarly, the European Court of Auditors report (European Court of Auditors, 2017) identified the most frequently employed customs duty evasion strategies, including the deliberate undervaluation of imported goods, misclassification by reassigning products to lower-duty categories, and the false declaration of origin. A study reported a notable disparity in tax and duty collection, which raising questions about the effectiveness of customs code implementation since the import duty was collected on only 47% of items imported through postal operators, compared with 99% for shipments handled by express carriers (Basalisco, 2016). Additionally, a study of Singapore's customs system revealed that 30% of declared submissions contained incorrect product codes (Ding

et al., 2015). For the national interest, the economic cost of significant misclassification is material and significantly undermines tax and duty compliance (Judge, 2020; Office of The Auditor General of Canada, 2017).

Customs authorities commonly adopt a risk-based lane system to optimise and streamline inspection procedures within the customs clearance process (Prost and Scattolo, 2024). The lane is categorised into green, yellow, or red lanes based on risk assessments derived from trader profiles, shipment histories, and declarations. The Green Lane is designated for low-risk packages and is typically cleared without inspection, as all documents comply with applicable legal standards. The Yellow Lane involves a documentary check, where the shipment's accompanying documentation is reviewed without a physical examination. Conversely, the Red Lane is designated for high-risk shipments and requires comprehensive physical inspections to verify the accuracy and legitimacy of the declared products. In the Green Lane, customs generally do not check declarations unless they are randomly selected or something triggers suspicion. The reason is to allocate resources efficiently, given the large volume of declarations in daily basis. However, the submission remains binding and can be reviewed at any time, including during post-clearance audits after product release (Grainger, 2024). Considering the large number of transactions processed daily, modern customs authorities often utilise machine-assisted tools to support customs officers in detecting anomalous transactions effectively (Kunickaite *et al.*, 2021).

2.4.1 Anomaly Detection using Heterogeneous Data

One of the earliest attempts to detect anomalies in customs fraud was performed in 2003 (A. Kumar Singh *et al.*, 2003), which introduced a fuzzy logic-based expert system as part of a DSS. The system used rule-based classification, derived from the experience of customs officers to flag transactions for further investigation based on HS classification. Another DSS study (Roman *et al.*, 2006) developed a system that identifies a transaction as an outlier if it exhibits an anomalous value in any dimension of the transaction category. In this study, an anomalous transaction was defined as one that deviates by more than three standard deviations from the average value of historical transactions. Correspondingly, a study used structured data to propose a hybrid framework to examine trade transactions for potential anomalies (Han and

Ireland, 2014). This method integrated manual, rule-based, and random selection methods, which uncovered that the inclusion of a random selection method was effective in deterring tax evasion by increasing unpredictability in the screening process.

A Bayesian network model was utilised in a study to detect anomalies in cargo trade documentation between exporting and importing parties for identifying potential smuggling from trade patterns (Triepels *et al.*, 2018). In a related study, Kim (Kim *et al.*, 2020) proposed a method for detecting customs fraud in trade transactions using a Gradient Boosting Decision Tree to classify illicitness and identify high-risk cases. In this study, both classification tasks and regression methods were employed to calculate potential collectable revenue from the transactions. Another line of work by Vanhoeyveld (Vanhoeyveld *et al.*, 2020) proposed a method to detect potential anomalies in customs data by combining a range of distinctive features from the trade transactions, which were categorised into behavioural data (i.e. commodity codes of all transactions in the same declaration) and high-cardinality data (i.e. the commodity code of a transaction). Both data types were then fed into classifiers consisting of the integration of SVM and the EasyEnsemble method to detect customs fraud. A different method to detect anomalies was performed involving concept drift in the trade data. The method was used to provide a signal by limiting the range of exploration and exploitation of a multi-armed bandit algorithm, used as an early system to detect potential anomalies or new fraud patterns that arise from transactions (Mai *et al.*, 2021). Another study was performed to compare various approaches to detect fraudulent behaviour using neural networks, fuzzy min-max classifiers, and logistic regression, where the data mainly relied on numerical attributes extracted from the trade declarations as model input (Kunickaite *et al.*, 2021).

Beyond tabular data, anomaly detection on image-based data types introduces distinct challenges given its reliance on high-dimensional visual representations. However, its potential has been explored for detecting smuggling activities, which utilised extracted X-ray cargo images to capture image features, such as shape and texture (Abdolshah *et al.*, 2017; Che *et al.*, 2018; Zhang *et al.*, 2014). In another research study, Zhang (Zhang *et al.*, 2014) proposed the use of joint shape and texture features extracted from X-ray images and provided them to an SVM classifier, which yielded better

classification performance. Similarly, X-ray cargo images were processed using Scale Invariant Feature Transforms, which were recognised to be invariant to several image transformations and to capture salient points of the image instead of using all parts of the image, for efficient and accurate output (Abdolshah *et al.*, 2017). Lastly, a subsequent research study applied deep neural networks as feature extractors of X-ray cargo images, used for an ensemble model for product classification (Che *et al.*, 2018). Using an imaging modality, their work demonstrated an improvement in automatically verifying commodities in the containers, involving a human-in-the-loop to justify the prediction outcome.

Taken all together, existing studies mentioned-above show that anomaly detection in the trade domain has employed heterogeneous types of data to tackle different forms of fraud. In the next section, the literature review will continue to explore text-based anomaly detection.

2.4.2 Anomaly Detection using Text Data

Anomalous text is implicit knowledge that is distinctively different from the general contextual idea (Hodge and Austin, 2004; Kannan *et al.*, 2017). The task to detect anomalous text refers to text anomaly detection, which the scope of input including documents or segments of text. This task faces some challenges inherent in the text, such as the nature of unstructured data with its inconsistency and morphological aspect (Chandola *et al.*, 2009; Mahapatra *et al.*, 2012). According to Taiye, text anomalies detection can manifest at different levels composed of five-tier hierarchy: topic, event, document, sentence, and word anomalies (Taiye, 2019). Product misclassification can be categorised as anomaly detection for document level where each of the description of the product is classified as a predefined product category or HS code. Identifying anomalies using text data can be examined across several linguistic aspects. Anomalous text which exhibits atypical vocabulary usage is referred to as a lexical anomaly, where in the case of product misclassification, some attributes of the product contain unrelated information to the primary information of the product. Semantic text anomaly refers to a text anomaly that differs from the general contextual meaning in the text; for this instance, the product description is typically misclassified into the wrong product category (Ait-Saada and Nadif, 2023; Hodge and Austin, 2004). However, semantic anomalies can be more complicated when a subtle meaning exists

between the product description and the category. In these circumstances, the contexts of the product belonging to others in the same category need to be considered.

Several studies have explored anomaly detection using text data; for instance, the study to detect product misclassification utilised a similarity-based method of the text and the hierarchical structure of the HS code (Spichakova and Haav, 2020b). Doc2vec was trained from product descriptions to generate non-contextualised word embeddings. The anomalies were calculated by computing the similarity of unseen transactions with the lexical representation of the product category and combining it with the score derived from the product code compared to the HS product taxonomy. However, this method might overlook the subtle variety of product descriptions to represent the product category and utilises non-contextualised embeddings, which offers limited capability in dealing with intricate product descriptions.

Another line of research by Chen utilised similarity-based anomaly detection to verify the correctness of product classification (Chen, Ben van Rijnsoever, *et al.*, 2021). The method created embeddings from historical trade transactions to produce a centroid vector representing each product category using pretrained sentence embeddings. Angular similarity was employed to measure the similarity between unseen transactions and the centroid vectors and to produce a computed score as an indicator to define potential anomalous transactions. However, the mean value used to represent the centroid vector of the normal product category might oversimplify the variety of product descriptions, which were later compared with unseen product transactions to detect anomalies.

Chan demonstrated the use of a two-step anomaly detection model for detecting customs fraud using trade data composed of product description and product price (Chan *et al.*, 2023). The method calculated the probability score of the product description given the product code submitted by traders. A subsequent step was then followed by assessing the normality of the product price compared to similar products in the same category. This method was operated under a supervised learning paradigm, where class labels were required to develop a model that was subsequently used to calculate the probability of the unseen product transaction. In addition, the price set as thresholds taken from the product transactions may not generalise well

across fine-grained product categories.

In a review study of anomaly detection using deep learning methods, Ruff (Ruff *et al.*, 2021) investigated various methods of anomaly detection, which were mainly divided into three major approaches: probabilistic, one-class classification, and reconstruction methods. It was found that reconstruction-based anomaly detection achieved the highest performance compared to other approaches evaluated using image synthetic MNIST-C and real-world MVTec-AD datasets. Reconstruction-based models, exemplified by autoencoders, have been adopted for anomaly detection due to their ability to identify anomalies (Chalapathy and Chawla, 2019; Pang *et al.*, 2022). This model is composed of two main components of a neural network, such as an encoder and a decoder. The encoder works to compress the input into a low-dimensional representation known as a latent or salient representation and subsequently to decode it to reconstruct the original input (Bank *et al.*, 2021; Hinton and Zemel, 1993). The model learns the underlying structure of latent representations through the compression and decompression processes that are subsequently important for reconstructing input data to detect anomalies. Anomaly detection with autoencoders has been effectively applied across various fields, including computer vision, cybersecurity, healthcare, manufacturing, biomedical imaging, signal processing, and business process event logs (Esmaeili *et al.*, 2023; Gorokhov *et al.*, 2023; Jia and Liu, 2023; Jiang *et al.*, 2023; Mohebbian *et al.*, 2021; Nawaz *et al.*, 2024; Nolle *et al.*, 2018).

2.4.3 Anomaly Detection with Autoencoder for Product Misclassification

A range of anomaly detection methods has been studied in the field of international trade for addressing tax and customs evasion. Previous studies have partially solved such activities by proposing anomaly detection methods relevant to data type, such as utilising X-ray scans for detecting smuggling (Abdolshah *et al.*, 2017; Zhang *et al.*, 2014), tabular numerical data to identify under- or overvalued transactions (Chan *et al.*, 2023; Vanhoeyveld *et al.*, 2020), and spatial network data to uncover trade shipment misrouting (Triepeles *et al.*, 2018). Although such studies have advanced efforts to combat different forms of tax evasion, current literature revealed that the detection of product misclassification in the trade domain is still underexplored and predominantly relies on supervised learning approaches (Chan *et al.*, 2023; Spichakova and Haav, 2020b).

The text element from trade declarations facilitates the potential to detect misclassification. This element is a key source of product information that must be matched with the assigned product category according to the HS codes guidance. From this view, product misclassification can be categorised as a form of anomaly when the product description is not aligned with the assigned product category. Nevertheless, detecting product misclassification in trade data remains a challenge, particularly for declared trade products that require fine-grained contextual understanding. Determining the valid product within the same product category becomes even more complex when ambiguous product descriptions exist in the transactions. Recent advancements in the NLP domain have enabled the generation of more representative and semantically rich text representations, which enhance the quality of text interpretation. This progress facilitates automated systems to comprehend text data, in particular product information. In such cases, misclassification can be detected from the presence of a deviated product description within the normal data distribution of product descriptions. In this context, employing semi-supervised or unsupervised learning paradigms enables the detection of anomalies using only product descriptions, where the availability of class labels (e.g., HS codes) is not necessary in the training phase. With this arrangement, previous discussions of autoencoders for anomaly detection is feasible to be implemented for detecting product misclassification. Autoencoders for anomaly detection depends on how they learn from the latent information of data to make the reconstruction-based mechanism functional.

Autoencoder architectures demonstrate the capability to learn latent representational characteristics of input data, which are subsequently utilised for data reconstruction processes. In essence, the learning of latent representations derived from normal data patterns facilitates the effective identification and distinction of anomalous data instances facilitated by a reconstruction-based mechanism. The literature suggests that the application of autoencoders to text data is still at an early stage and has yet to be fully examined. Consequently, their potential for detecting anomalies using text data composed of mixed attributes, such as product description, has not been comprehensively investigated (Li, Pei, *et al.*, 2023; Novoa-Paradela *et al.*, 2024). The input data for autoencoders network are suggested to use contextualised embeddings derived from pretrained language models to enhance model performance. Providing

semantically rich representations enables the autoencoder to detect anomalies more effectively, as the inputs already capture meaningful contextual information. Furthermore, with the availability of diverse autoencoder architectures, this thesis aims to develop text anomaly detection models and perform a comparative evaluation to identify the variant that demonstrates the most promising empirical performance for downstream applications.

2.5 Bilingual Classification in Trade Transactions

In the previous section, we focused on detecting anomalies within trade transactions, while in this section, we present another challenge in the trade domain that is the presence of more than one language within product descriptions (de Artiñano *et al.*, 2023; de Lima *et al.*, 2022; Paramartha *et al.*, 2021). The section begins by reviewing studies that address product classification using monolingual data, and then proceeds to bilingual classification, followed by cross-lingual classification, and concludes with the identified gaps and proposed research objectives.

Trade declaration submissions typically require product descriptions to be written in the official language mandated by local authorities. Nevertheless, due to the diverse linguistic backgrounds of users, product descriptions are often written in different languages. For example, in the case study on Indonesian trade product classification (Paramartha *et al.*, 2021), traders are generally expected to describe products in the Indonesian language. However, it is common to find mixed-language descriptions, mainly in English terms, used to convey equivalent or technical meanings of the product. This practice is, nevertheless, permitted under Indonesian domestic trade policies to write in English for the completeness and technical clarity of product descriptions (Indonesian Customs, 2009). Consequently, in practical implementation, it requires more effort for the authority to understand the product description, which has similar categories described in different languages during document inspections. A similar phenomenon is also observed in the Chilean trade dataset (Marra de Artiñano *et al.*, 2023), where Spanish terminology is frequently used to replace or complement equivalent terms in other languages within product descriptions.

2.5.1 Monolingual Classification

The existing literature on automating product classification suggests that monolingual

approaches are primarily used as a benchmark to evaluate model performance. Most studies utilised an English dataset from a publicly available source, such as the U.S. import trade data employed in several studies (Luppens, 2019; Ruder, 2020; Shubham *et al.*, 2023; Spichakova and Haav, 2020a). Other studies utilise datasets specific to particular jurisdictions, such as product classification using Chinese datasets (Liao *et al.*, 2024), Brazilian datasets (de Lima *et al.*, 2022), Chilean datasets (Marra de Artiñano *et al.*, 2023), Korean datasets (Lee *et al.*, 2024), and Indonesian datasets (Paramartha *et al.*, 2021). However, such specific jurisdiction datasets are not publicly available due to a non-disclosure agreement, which limits access and comparability studies. In several studies that perform classification tasks using non-English datasets, the model is generally trained with multilingual pretrained models, such as the predictive model for Brazilian datasets (de Lima *et al.*, 2022) and Chilean datasets (de Artiñano *et al.*, 2023) that use multilingual BERT (mBERT). However, some studies utilise modern language models that are specifically trained on analogous training data. For example, research employed Chinese trade datasets utilising ERNIE (Liao *et al.*, 2024) and a study performed on Korean datasets using KoBERT (Lee *et al.*, 2024). These implementations showed that multilingual pretrained language models or specific language model analogues with training data frequently become a choice for the text representation in dealing with non-English trade datasets.

2.5.2 Bilingual Classification

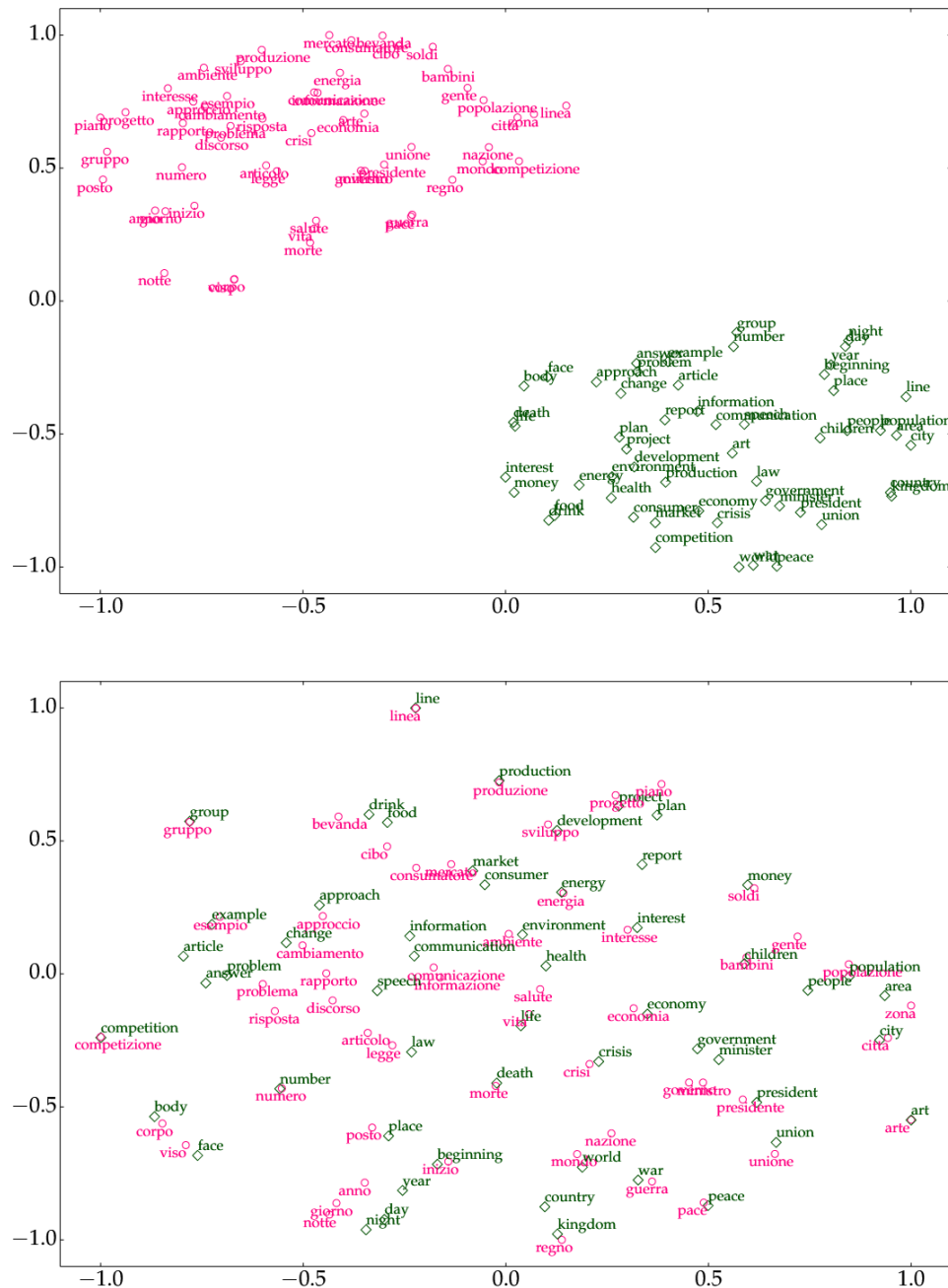
Prior research primarily centres on monolingual language, leaving bilingual language underexplored. However, outside of the international trade domain, research studies were performed on utilising bilingual datasets in various domains. The first study, which relates to product classification, was conducted to classify the brand of a product in bilingual languages, English and Russian (Kulagin *et al.*, 2019). The traditional method was carried out using a range of text encoders, including TF-IDF and LSTM and traditional machine learning algorithms as classifiers, including SVM and kNN, with the result determining that CNN represents the most effective model. Additionally, a previous study employed a classification task for categorising products, which incorporated two languages, Spanish and French (Sharma and Parwekar, 2023), revealing that the DT achieved the highest accuracy compared to the XGBoost or Gradient Boosting classifier. All things considered, both studies trained the models on

bilingual languages using a similar method that employed a joint training approach by combining two datasets used in the training phase.

Several studies have conducted experiments using contextualised embeddings generated by transformer-based models for bilingual classification. A research study used Turkish and English datasets to classify grocery product title categories to fine-tune pretrained multilingual models, such as mBERT and XLM-RoBERTa (Jahanshahi *et al.*, 2021). The empirical results found that the implementation of this approach generated better performance compared to baseline models that used non-contextualised embeddings combined with traditional classifiers, such as SVM and XGBoost. Another implementation of pretrained multilingual language models employed a stacking ensemble that combined predicted probabilities from DeBERTa and InfoXLM models with handcrafted features and query locale information as inputs to a LightGBM meta-learner to predict e-commerce product classification (Zhang *et al.*, 2022).

2.5.3 Cross-lingual Embeddings

Bilingual classification encounters representational challenges due to morphological, syntactic, and semantic variations between languages. A key objective is enabling models to align cross-lingual semantics within the embedding space, ensuring that words with similar meanings across languages are positioned closely together. Cross-lingual word embeddings accommodate the transfer of lexical knowledge across languages by representing words from multiple languages in a shared embedding space (Ruder *et al.*, 2019). Employing a cross-lingual method enables the alignment of word representations from equivalent text data of bilingual languages, resulting in the embeddings being mapped in similar locations in the embedding space, as shown in Figure 2.6. This approach offers benefits for cross-lingual document classification and information retrieval across languages (Ruder *et al.*, 2019).



with translations in other languages, demonstrated in the Language-Agnostic Sentence Representations (LASER) model (Artetxe and Schwenk, 2019). LASER employs an encoder-decoder architecture to generate ready-to-use cross-lingual embeddings that work across the languages it was trained on. The Cross-lingual Language Model (XLM) utilises a Transformer-based architecture for cross-language representation learning through contextual representations rather than word-level translation dictionaries and facilitates fine-tuning for domain-specific applications (Conneau and Lample, 2019).

Another training approach used to align the embeddings is the joint training method. The basic idea of this method is to train the model on multilingual data simultaneously using shared vocabularies with an example of language models generated by this method is multilingual BERT (Devlin *et al.*, 2019). An extensive multilingual corpus is needed to use this method to promote an implicit training strategy that facilitates alignment by employing a shared subword vocabulary across languages. The process employs architecture based on Transformer to capture semantic representations across different languages and enable effective cross-lingual transfer. XLM-RoBERTa (XLM-R) (Conneau *et al.*, 2020a) is a subsequent study extending this method by pretraining on substantially larger corpora. The availability of multilingual pretrained language models, such as mBERT and XLM-R, has leveraged the improvement the multilingual tasks, which in practice have been adopted into specific domains through fine-tuning approaches (de Artiñano *et al.*, 2023; Jahanshahi *et al.*, 2021; de Lima *et al.*, 2022), and in the process involving knowledge distillation (Park *et al.*, 2024). This experiment was conducted by designing XLM-R as the teacher model to guide a student model in aligning source language embeddings with the target language. Multilingual models also offer the capacity to transfer knowledge from high-resource to low-resource languages. A study examined the task of predicting similar products across online marketplaces, employing English as the high-resource language and German as the low-resource language (Peeters and Bizer, 2022). In this investigation, incremental addition of training samples was employed to facilitate implicit alignment, with the results showing that mBERT outperformed XLM-R.

In our endeavour to address a low-resource trade data challenge, analysing the characteristics of trade datasets shows they are clearly dominated by domain-specific

product terminology, which is generally used for trade exchange. Another notable specification innate in the product description is the brevity of the text length, identified as short text, which often yields insufficient features and impacts the model, resulting in poor classification performance (Chuanakrud *et al.*, 2021; Tang *et al.*, 2022). For text classification tasks using short texts, few words often define the model outcomes (Zhou *et al.*, 2023). Given these aspects, involving keywords of the main product description is potentially supportive in representing product information in a better way for classification tasks. A keyword is defined as a word or phrase that encapsulates the main topic, theme, or essential information of a text, which often functions as a discriminative marker enhancing classification performance by guiding models to emphasize critical features (Blanchard *et al.*, 2022; Chuanakrud *et al.*, 2021). Data augmentation involving keywords are performed at the lexical level by replacing the keywords in text with their similar words or by selecting substitute words based on proximity in embedding spaces to preserve semantic similarity (Bayer *et al.*, 2023). The role of keywords could also be expanded into category labels, which were incorporated with text using a keyword-text graph network as features to feed into a graph convolutional network, as performed in the study of Chuanakrud (Chuanakrud *et al.*, 2021). Keywords have been utilised within data augmentation strategies, with techniques involving the replacement of specific keywords or key terms within text samples with contextually suitable alternatives (Xiang *et al.*, 2020). In general, data augmentation itself is particularly advantageous in creating sample data to handle low-resource contexts, which has evidently improved model performance (Nair *et al.*, 2024). Transactions containing terms in the domain of trade have complex meanings and additional attributes related to the main product. For this reason, keywords can be valuable for narrowing down the focus of the product description to the core meaning.

2.5.4 Bilingual Classification with Keywords Integration

The existence of bilingual languages within trade data uniquely challenges application tasks for automatically classifying product categories. While a fine-tuning multilingual approach is regarded as improving the accuracy of the model, this approach requires sufficient bilingual training datasets. The limited number of trade datasets, especially non-English datasets in the public domain, becomes a challenge for research endeavours. While most studies utilising non-English datasets focus on developing

product classification tasks specific to a country's jurisdiction (de Artiñano *et al.*, 2023; Liao *et al.*, 2024, 2024; de Lima *et al.*, 2022; Paramartha *et al.*, 2021), the accuracy performance of these monolingual models is expected to be suboptimal when presented with unseen trade data in languages absent from the training phase.

Several studies outside the domain of international trade have proposed methods for classification using bilingual datasets (Jahanshahi *et al.*, 2021; Kulagin *et al.*, 2019; Sharma and Parwekar, 2023; Zhang *et al.*, 2022). When bilingual datasets are available, they are often combined to train machine learning algorithms, with input representations generated using various encoding methods. However, these approaches have drawbacks in their ability to capture textual context, leading to suboptimal model performance. On the other hand, an endeavour to utilise modern language models, including transformer-based pretrained models, seems promising to capture the nuances of the bilingual model to achieve optimal performance. However, to achieve an optimal performance, a fine-tuning approach becomes a well-established strategy to achieve good performance; the process itself may consist of pairing the training samples in the bilingual setting and joint sample training. However, this approach faces a concern related to the limited availability of the target trade dataset. The presence of multiple languages in trade descriptions presents a significant challenge for training classification systems. Consequently, the development of bilingual models must consider the limited availability of target datasets in the public domain.

Cross-lingual embedding is necessary in the context of limited resources of trade data, as it enables the transfer of knowledge from high-resource to low-resource languages, with English being considered as a high-resource dataset due to its availability in the public domain. This study proposes a novel approach to fine-tune the model and leverage modern pretrained language models by incorporating keywords to serve as an auxiliary method to improve classification. The observation that utilising keywords in classification tasks was adopted as part of data augmentation techniques, where they are used to replace similar words to generate additional samples for model training, resulting in performance gains for classification tasks (Nair *et al.*, 2024; Xiang *et al.*, 2020). However, there remains a limited understanding of how utilising these keywords beyond previous approaches can be effectively applied to bilingual

classification tasks. The limited discussion in the literature on bilingual trade transactions presents an opportunity for developing models that can fulfil this necessity and address low-resource multilingual trade settings.

2.6 Summary

This chapter reviews the existing literature that forms the foundation of this research, focusing on developmental approaches and applications through a range of machine learning tasks, including neural networks that utilise international trade data. At the beginning of the section, an overview of the trade codification system (HS codes) is presented, complemented by trade declarations used in international trade. Various trade datasets have been utilised in research studies, but the limited availability of datasets in the public domain highlights a real issue in this domain. The review proceeds with an examination of a classification pipeline. It covers the primary stages performed for text classification task using product declarations. These stages include explanations of text classification approaches, ranging from preprocessing techniques, text representation methods, and classification approaches. The discussion underscores the promise of transformer-based language models for improving model performance. The coverage of product classification is then further extended to investigate sentence embeddings to represent product declarations. Reviewing the literature in this area underlines gaps to be addressed that are expected to improve trade classification performance. The subsequent section introduces anomaly detection methods applied within the trade domain, focusing on product misclassification, and provides an overview of approaches tailored to traded product data. Particular attention is given to the use of autoencoder networks, which are intended as a method for detecting anomalous transactions in traded products. The final part of the discussion reviews existing research on bilingual classification in trade data. The presence of multiple languages in traded product declarations underlines the underlying complexities of product classification, a challenge further compounded by the scarcity of datasets in low-resource languages. Incorporating keywords as an auxiliary feature to improve cross-lingual embeddings offers a promising method to improve bilingual classification performance.

The subsequent chapter will address the first objective of this thesis in developing a text classification pipeline designed for trade product classification.

Chapter 3

Optimising the Text Classification Pipeline for Trade Product Classification

This chapter addresses the first research objective, which involves the development of classification models to perform text-based classification of international trade product declarations. The methodology employed in this thesis implements a systematic framework used for text classification pipelines, which is designed to address the unique characteristics and challenges associated with domain-specific trade data. The methodology employs three primary strategies: configuring an optimal strategy for text preprocessing techniques, optimising feature extraction embeddings through fine-tuning language models, and applying classification performance using two distinct classification approaches. The effectiveness of the proposed methods is demonstrated by improved classification accuracy in predicting product categories, which was validated on a publicly available trade dataset and compared with baseline models and previous studies that utilised a wide range of product categories.

3.1 Introduction

Trade declaration submission is a part of procedural standards in international trade for product exchange, which involves assigning product descriptions to a valid product code (HS code) (World Customs Organization, 2018b). Several studies have developed methods to assist traders in managing these necessities to support traders in ensuring compliance with international standards and reducing the risk of negative drawbacks, including penalties or delays in the trade process.

Driven by advances in technology, research on text classification for product

categorisation continues to evolve; this also reflects the persistent challenges associated with the performance of classification methods developed for handling this issue (Brigstock, 2017; Heijde, 2019; Lee *et al.*, 2024). The terminology gaps, which present a gap in understanding between the usage and definition of the products, become a key factor contributing to classification challenges. In practice, the terminology variations in industry-specific language often led to the use of different terms for similar products, which in turn creates ambiguity and inconsistency in aligning product descriptions with standardised nomenclature and codification systems (Barbosa, 2021; Ding *et al.*, 2015; Heijde, 2019; Kappler, 2011). To illustrate the problem, take the example of a trade product such as an "*MP3 player*," a term widely used by traders to describe an electronic device used to play music. However, under the HS nomenclature, it falls within the broader category of "*Sound recording or reproducing apparatus: using magnetic, optical or semiconductor media*," which presents an uncommon and complex definition that requires effort from traders to understand the official regulatory terms. Moreover, the evolving technological features of products often lead to ambiguity in classification, as traders may struggle to interpret the appropriate category. For instance, a "*smartwatch*" may be intuitively classified as either a watch or a mobile device, given its multifunctional capabilities and digital features (Lee *et al.*, 2024). Some of the inherent complexities mentioned above become a reason to motivate this thesis in mitigating the ongoing issue, especially when classifying products into a valid code category. The implementation of automated prediction tools presents a valuable opportunity to enhance consistency and reduce subjective interpretation when classifying products into HS code categories. Nonetheless, the limited availability of such tools and their still limited functionality strengthen our motivation to develop solutions using recent technological advancements for practical effectiveness.

In the machine learning domain, automatic product classification is regarded as a text classification problem (Aggarwal and Zhai, 2012; Kowsari *et al.*, 2019). It typically involves several key steps, including preliminary tasks, such as text cleaning or preprocessing, feature extraction, and classification process (Jo, 2018; Kowsari *et al.*, 2019). Product descriptions are characterised by the presence of supplementary information to the main product, the product attributes such as brand names, product specifications, and the product's units of measurement. These attributes are

commonly expressed through a combination of alphabetic characters, punctuation, numerical values, and alphanumeric sequences, all embedded within a typically brief text format (Ding *et al.*, 2015; Zhou *et al.*, 2023). Numerous studies in trade classification have adopted a range of preprocessing techniques intended to enhance text representation for downstream classification tasks, as outlined in Chapter 2. Nevertheless, a structured or standardised framework for selecting appropriate preprocessing methods remains absent, and the specific contribution of each technique to a model classification performance is still unclear.

Text preprocessing serves as a necessary intervention to address the heterogeneous nature of the mixed product attributes. As highlighted in prior survey studies (Kadhim, 2019; Kowsari *et al.*, 2019; Li, Peng, *et al.*, 2022), preprocessing represents a foundational step in text classification, involving several transformation processes to standardise text data for trade transactions. The shortness of product descriptions often results in insufficient feature representation, limiting the model performance (Tang *et al.*, 2022). Therefore, the application of preprocessing techniques for text data should be taken carefully to evaluate their impact on classification outcomes. The limited attention given to preprocessing in the existing literature, particularly in utilising large language models as text encoders, highlights the need to investigate whether preprocessing strategies significantly impact the model's predictive capabilities (Li, Peng, *et al.*, 2022). Text preprocessing has an important role in NLP performance, but it has often been overlooked in deep learning research because the focus has shifted toward model architecture and training techniques (Camacho-Collados and Pilehvar, 2018; Naseem, Razzak, Khan, *et al.*, 2021). Therefore, an evaluation of preprocessing strategies is important to assess their impact on trade classification performance.

A suitable text encoder is influential in generating effective representations of raw textual input. To illustrate this necessity, consider the word "*Apple*" in two different contexts: "*Fuji Apple Class-2 ...*" and "*Spigen Apple (42mm) ...*". Distinguishing the first as a fruit and the second as a brand of electronic device may already pose a challenge for human readers, let alone for a machine learning classifier that relies on numerical input features. However, most studies in the domain of trade product classification rely on traditional text encoders, such as OHE (Barbosa, 2021), BoW (Harsani *et al.*, 2020), TF-IDF (Altaheri and Shaalan, 2020; Paramartha *et al.*, 2021;

Ryzhova and Sochenkov, 2019), or non-contextualised embeddings like Word2Vec (Luppess, 2019; Ruder, 2020; Ryzhova and Sochenkov, 2019), Doc2vec (Spichakova and Haav, 2020a), and GloVe (Ruder, 2020), which may limit the model's ability to capture semantic nuances. The application of contextualised word embeddings, especially those generated through transformer-based text encoders, is still relatively limited in the domain of trade product classification (de Lima *et al.*, 2022).

Transformer-based models have demonstrated effectiveness in generating word embeddings to enhance text representation and capture the surrounding context to facilitate resolving ambiguity. This contextualised word representation is also important for text understanding and offers advantages for improving model classification performance (Garrido-Merchan *et al.*, 2023). However, a concern with pretrained language models is that these models were trained on general-domain corpora, which may not adequately capture the specialised terminology and context found in specific domains, such as trade data. Additionally, different pretrained models often show varying levels of effectiveness on the same downstream task due to differences in their pretraining objectives and model architectures (Qiu *et al.*, 2020). With the growing number of available pretrained language models hosted on online model hubs (Wolf *et al.*, 2020), determining the most appropriate language model for trade-specific datasets remains a question. Therefore, it is essential to identify pretrained language models best suited for trade domain classification tasks and fine-tune them using trade-specific datasets. Fine-tuning method represents a key component of transfer learning methodologies, enabling the models to adapt their learned representations to domain-specific requirements (Pan and Yang, 2010). Pretrained language models possess extensive linguistic knowledge acquired from the pretraining process on large-scale corpora; therefore, they demonstrate strong generalisation capabilities even when fine-tuned on limited or imbalanced datasets typical of specialised domains (Arslan and Cruz, 2024; Devlin *et al.*, 2019).

Another important consideration in this thesis is the selection of classification approaches. Most studies in the international trade product classification use flat classification (Altaheri and Shaalan, 2020; Harsani *et al.*, 2020; Paramartha *et al.*, 2021; Ruder, 2020; Ryzhova and Sochenkov, 2019; Spichakova and Haav, 2020a), which are considered ineffective when dealing with a large number of product

categories and imbalanced data (Cotacallapa *et al.*, 2024; Silla and Freitas, 2011). On the other hand, hierarchical classification methods employ various strategies for leveraging taxonomic structure, ranging from local node-specific classifiers to global approaches that consider the entire hierarchy simultaneously. However, employing this approach presents several challenges, including increased model complexity, evaluation difficulties, and computational overhead in hierarchical processing (Silla and Freitas, 2011). Despite the limited implementation of hierarchical classification studies in the trade domain (Barbosa, 2021; Shubham *et al.*, 2023), to our knowledge, no existing research has systematically compared the effectiveness of flat versus hierarchical classification approaches using fine-tuned pretrained language models. Consequently, the most suitable classification method for trade applications has yet to be determined.

Performing a comparative study on two classification approaches using a trade dataset is important to determine which one can achieve optimal performance. This research explores the implementation of both flat and hierarchical classification methods by considering the capability of modern pretrained language models for understanding the text data, which is expected to generate a strong performance result compared to the traditional ones. Our purpose in this thesis aligns with previous study recommendations (Cotacallapa *et al.*, 2024), which compared flat and hierarchical classification methods using e-commercial data for product classification, despite of their work relied fully on traditional machine learning algorithms.

This thesis aims to develop a text classification pipeline incorporating three key components, summarised as the application of text preprocessing strategies, a fine-tuned method for pretrained language models, and a classification approach. The hypothesis posits that these integrated approaches will provide a more integrated solution for improving classification performance for trade product classification. The thesis employed the methodology with comprehensive experiments utilising publicly available datasets and comparison against baseline methods.

The overall contribution of this study is outlined as follows:

1. We propose a systematic text preprocessing strategy to quantify the impact on performance of product classification using trade data by identifying optimal

preprocessing strategies, which providing support for potential applicability in various domains that have similar data characteristics.

2. This research proposes fine-tuning methods on transformer-based language models using a trade dataset, comparing their performances and benchmarking them against baseline methods. The findings identify which transformer-based language model provides the most suitable representations for trade-related classification tasks.
3. This thesis proposes a novel hierarchical classification approach to achieve the best results for trade product categorisation. The classification pipeline produces models that outperform baseline methods in the literature and presents potential for deployment in automated trade classification systems.

This chapter is outlined as follows. Section 3.2 introduces the methodology for text classification for trade product classification and details the experimental design. Section 3.3 reports the experimental results with discussion and interpretation of the findings. Section 3.4 provides an overview of the research and conclusions.

3.2 Methodology

The proposed methodology comprises three sequential main components and employs a comparative experimental design to identify the optimal configuration at each phase. The first step is to implement text preprocessing strategies to identify the most effective techniques for handling trade product descriptions. The second step is to select a pretrained language model as the text encoder, which is subsequently fine-tuned on the domain-specific dataset. In the final step, the generated embeddings are utilised as input features for two types of classifiers: flat and hierarchical classification approaches. In the final phase, an evaluation of the models' performances identifies the most effective combination for optimising trade product classification. An overview of the entire pipeline is presented in Figure 3.1.

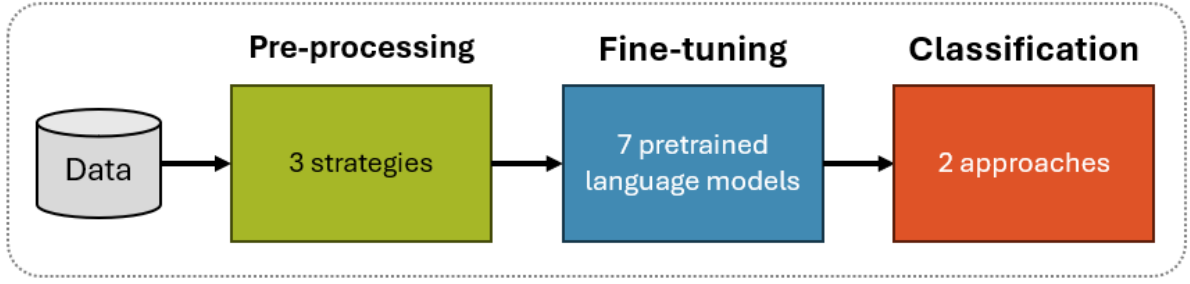


Figure 3.1 Overview of the classification pipeline comprising data pre-processing strategies, fine-tuning pretrained language models, and classification approaches.

3.2.1 Problem Formulation

This section exhibits a formulation of the problem, forming the proposed methodology. The dataset consists of pairs, $D = \{(s_i, y_i)\}_{i=1}^n$, where s_i denotes a product description and y_i represents its associated label from a set of C categories, with $y_i \in \{1 \dots C\}$. Preprocessing strategies are applied to each s_i , and will be detailed in section 3.2.3. A selection of pretrained language models used for this study is outlined in Section 3.2.4. These text encoders are based on a Transformer-based architecture (Vaswani *et al.*, 2017) and fine-tuned end-to-end on dataset D using supervised learning approach. The product declarations are represented by an aggregate of contextualised word embeddings extracted from the final hidden layer of the encoder network, $h_{cls} \in \mathbb{R}^d$, where d is the dimensionality of the embeddings. The classification head is placed on top of this text representation as a FFN with fully connected layers and non-linear activation functions. Two approaches are designed to classify the input data: flat classification and hierarchical classification, which will be explained in section 3.2.5.

3.2.2 Dataset

Among the limited publicly available sources of international trade datasets, a trade dataset is provided by Enigma, accessible via a public repository¹³ (Enigma, 2018). This dataset has been widely utilised in multiple studies of classification of product codes (Luppens, 2019; Ruder, 2020; Shubham *et al.*, 2023; Spichakova and Haav, 2020a). The dataset was obtained from U.S. Customs and Border Protection (CBP),

¹³ <https://aws.amazon.com/marketplace>

containing detailed records of international trade from 2018 to 2020. The data was separated in several raw text files, including cargo descriptions, tariff codes, shipper information, carrier details, and other related files. The dataset was prepared, including a cleaning and merging process to confirm coherence and usability for subsequent stages. All available data was used as-is, under the assumption of correctness, although the possibility of misclassification may be present. Relevant parameters were used to consolidate multiple files as described in the page data repository, resulting in a single source containing product descriptions and associated HS codes. After removing duplicate entries, the final dataset comprises 12,391,206 transactions. The dataset was subsequently filtered to ensure that all entries contain a mandatory 6-digit HS code. HS codes with fewer than six digits were excluded, while those exceeding six digits were truncated to retain only the first six digits. After dropping unnecessary columns and removing data duplication, the final statistics of the dataset used for this study are presented in Table 3.1. The dataset was split into 70% for training and 30% for testing to balance computation and still be statistically reliable, given the large sample size.

Year	Number of transactions
2018	491,405
2019	534,186
2020	397,374
Total rows	1,422,965

Table 3.1 The trade dataset statistics.

3.2.3 Data Preprocessing

Trade product descriptions contain various product attributes and elements, which can obscure the primary product, including punctuation marks, numerical codes, alphanumeric identifiers, and supplementary product information.

A preprocessing strategy is adopted to evaluate if text preprocessing has a significant impact on model performance, in particular when language models are employed for extracting text features. Accordingly, this research aims to explore which product attributes convey semantic or contextual information that improves text classification

performance. The choice of text preprocessing techniques was adopted from previous studies (Altaheri and Shaalan, 2020; Amel *et al.*, 2023; Chen, Bromuri, *et al.*, 2021; Ding *et al.*, 2015; Harsani *et al.*, 2020; Heijde, 2019; Li and Li, 2019; de Lima *et al.*, 2022; Marra de Artiñano *et al.*, 2023; Paramartha *et al.*, 2021; Ruder, 2020; Shubham *et al.*, 2023; Stassin *et al.*, 2023). We did not use all of the techniques; instead, we selected some techniques to balance between removing irrelevant noise and retaining informative features critical for the model to learn. We divided the preprocessing steps into two main steps: transforming the text and removing noisy or irrelevant attributes. Text transformation includes converting all characters to lowercase, a commonly adopted step in text preprocessing. Next, the elimination of selected product attributes is divided into three categories, as presented in Table 3.2. First, punctuation marks were systematically removed as they contribute minimal semantic value to categorisation tasks and have given an advantage to reduce noise. Second, numerical values and alphanumeric data were filtered out because of their inconsistent semantic contribution, as previous studies demonstrate that numerical specifications often represent variable product attributes instead of core product information. Finally, domain-specific words, such as unit measurements and technical product specifications, were eliminated to supply the model with important product information.

Preprocessing technique	Example
Removing punctuation marks	. , ; : ? ! " ' () [] { } - _ / \ & % \$ # @ * + = ~ ` ^
Removing numbers and words mixed with numbers	1, 123, hku456, 3gb
Removing additional product information	no, pcs, kg, cm, qty

Table 3.2 Preprocessing techniques used for trade data.

The text preprocessing was configured into three strategies, where each strategy applies preprocessing steps that were built incrementally on the previous strategy, as presented in Table 3.3. The first strategy, preprocessing 1 (PP-1), utilised basic text transformations, including lowercasing and removing punctuation marks. Preprocessing 2 (PP-2) built upon this by additionally removing numerical values, including words that contain alphanumeric combinations. Finally, preprocessing 3 (PP-

3) combined the previous techniques and additionally removes predefined supplementary product attributes, which are commonly found in general product descriptions.

PP version	Preprocessing Steps
PP-1	• Lowercase • Removing punctuation
PP-2	• Lowercase • Removing punctuation • Removing numbers and words mixed with numbers
PP-3	• Lowercase • Removing punctuation • Removing numbers and words mixed with numbers • Removing additional product information.

Table 3.3 List of preprocessing strategies and techniques applied to trade transaction data.

Table 3.4 summarises the dataset statistics after each preprocessing strategy was applied. A progressive decrease in the number of words is anticipated, assuming the cumulative removal of selected elements in each strategy. However, the number of samples and class labels was reduced insignificantly across all text preprocessing strategies.

Statistic Attribute	PP-1	PP-2	PP-3
Number of rows	1,421,962	1,399,574	1,386,330
Number of product categories (HS-6 codes)	3,566	3,541	3,525
Number of words	19,461,401	13,858,411	9,997,026
Mean of words	13	9	7
Maximum of words	96	66	64

Table 3.4 Dataset statistics across applied preprocessing strategies.

3.2.4 Pretrained Language Models

Following the preprocessing stage, the dataset was processed using pretrained

transformer-based language models to generate contextual embeddings from the text data. As outlined in the initial methodology of this study, the focus is on pretrained models built upon the Transformer architecture (Vaswani *et al.*, 2017). We selected Transformer-based language models with an architecture that only utilises the encoder component, which are designed specially for language understanding and well-suited for classification tasks (Benayas *et al.*, 2024; Chadha, 2020; Devlin *et al.*, 2019).

A large number of pretrained language models have been developed and widely utilised to solve a variety of language-related tasks and have been made publicly available (Qiu *et al.*, 2020; Wolf *et al.*, 2020). Therefore, determining pretrained language models that present strong performance on the target task is essential because different models often affect different results due to differences in their pretraining objectives, model architectures, and training corpora (Qiu *et al.*, 2020). We focused on selecting several models that have been previously evaluated and shown to achieve strong or state-of-the-art results for text classification tasks, as evidenced by benchmark leaderboards¹⁴. The selected pretrained language models used in the following experiments are as follows:

1. ALBERT (Lan *et al.*, 2020)
2. BERT (Devlin *et al.*, 2019)
3. DeBERTa (He *et al.*, 2023)
4. DistilBERT (Sanh *et al.*, 2020)
5. ELECTRA (Clark *et al.*, 2020)
6. RoBERTa (Liu *et al.*, 2019)
7. XL-NET (Yang *et al.*, 2019)

A pretrained language model as the base encoder is required to use the tokenizer dedicated to that model. This tokenizer breaks down text into smaller components known as tokens. Character-level tokenisation breaks input text into individual characters, which can be beneficial for addressing misspellings and rare words, though it is computationally and memory-intensive (Zhang *et al.*, 2016). Word-level tokenisation relies on whitespace to split text input. However, this approach introduces several issues, such as handling misspellings and out-of-vocabulary (Sennrich *et al.*,

¹⁴ <https://paperswithcode.com/task/text-classification>

2016). Incorporating these variations causes vocabulary to expand rapidly, hindering model efficiency. Subword tokenization, such as Byte Pair Encoding (BPE) (Sennrich *et al.*, 2016), WordPiece (Schuster and Nakajima, 2012), SentencePiece (Kudo and Richardson, 2018), represents the dominant tokenisation approach in modern pretrained language models, as shown in Table 3.5. This method arranges a balance between character and word-level approaches, handling rare words and morphological variations while maintaining manageable vocabulary sizes.

Model	Tokenization Method	Number of vocabularies
ALBERT	SentencePiece	30,000
BERT	WordPiece	30,522
DeBERTa	SentencePiece	50,265
DistilBERT	WordPiece	30,522
ELECTRA	WordPiece	30,522
RoBERTa	Byte Pair Encoding	50,265
XLNet	SentencePiece	32,000

Table 3.5 Selected transformer-based models with their corresponding tokenisation algorithms and vocabulary sizes.

Transformer-based models adopt a hybrid tokenisation strategy by integrating the advantages of both character-level and word-level tokenisation techniques, segmenting input text into words, subwords, or characters, followed by numericalisation, mapping each token to a unique integer from a predefined vocabulary. This vocabulary is learned from extensive corpora using tokenisation algorithms and is specific to each model. The tokenisation outcome is influenced by the algorithm and configuration of the tokenizer. Although some models implement similar tokenisation algorithms (ALBERT, DeBERTa, and XLNet), they generate different token outputs, which is caused by variations of pretraining data and vocabulary construction. Tokenisation results varied across models in both token count and structure, as presented in Table 3.6. These differences can impact the model performance because they affect the granularity of text segmentation and the representations of generated embeddings.

Model	Tokens	Unique Count
<i>Example</i>	<i>bravox loudspeakersas follow malibu mb12 v2 malibu mb8 g10xsbss revo 8xswss axis12 axis 65 recon 6bg revo 12 hpbs4 v2 g10xss</i>	-
ALBERT	['_bravo', 'x', '_loudspeaker', 's', 'as', '_follow', '_malibu', '_mb', '12', '_v', '2', '_malibu', '_mb', '8', '_g', '10', 'x', 's', 'bs', 's', '_rev', 'o', '_8', 'x', 's', 'ws', 's', '_axis', '12', '_axis', '_65', '_recon', '_6', 'bg', '_rev', 'o', '_12', '_hp', 'bs', '4', '_v', '2', '_g', '10', 'x', 's', 's']	47
BERT	['bravo', '##x', 'loud', '##sp', '##eak', '##ers', '##as', 'follow', 'malibu', 'mb', '##12', 'v', '##2', 'malibu', 'mb', '##8', 'g', '##10', '##x', '##sb', '##ss', 'rev', '##o', '8', '##x', '##sw', '##ss', 'axis', '##12', 'axis', '65', 'rec', '##on', '6', '##b', '##g', 'rev', '##o', '12', 'hp', '##bs', '##4', 'v', '##2', 'g', '##10', '##x', '##ss']	48
DeBERTa	['_bravo', 'x', '_loudspeakers', 'as', '_follow', '_malibu', '_mb', '12', '_v', '2', '_malibu', '_mb', '8', '_g', '10', 'x', 's', 'b', 's', 's', '_re', 'vo', '_8', 'x', 's', 'ws', 's', '_axis', '12', '_axis', '_65', '_recon', '_6', 'bg', '_re', 'vo', '_12', '_hp', 'b', 's', '4', '_v', '2', '_g', '10', 'x', 's', 's']	48
DistilBERT	['bravo', '##x', 'loud', '##sp', '##eak', '##ers', '##as', 'follow', 'malibu', 'mb', '##12', 'v', '##2', 'malibu', 'mb', '##8', 'g', '##10', '##x', '##sb', '##ss', 'rev', '##o', '8', '##x', '##sw', '##ss', 'axis', '##12', 'axis', '65', 'rec', '##on', '6', '##b', '##g', 'rev', '##o', '12', 'hp', '##bs', '##4', 'v', '##2', 'g', '##10', '##x', '##ss']	48
ELECTRA	['bravo', '##x', 'loud', '##sp', '##eak', '##ers', '##as', 'follow', 'malibu', 'mb', '##12', 'v', '##2', 'malibu', 'mb', '##8', 'g', '##10', '##x', '##sb', '##ss', 'rev', '##o', '8', '##x', '##sw', '##ss', 'axis', '##12', 'axis', '65', 'rec', '##on', '6', '##b', '##g', 'rev', '##o', '12', 'hp', '##bs', '##4', 'v', '##2', 'g', '##10', '##x', '##ss']	48
RoBERTa	['b', 'rav', 'ox', 'Ġloudspe', 'akers', 'as', 'Ġfollow', 'Ġmal', 'ibu', 'Ġm', 'b', '12', 'Ġv', '2', 'Ġmal', 'ibu', 'Ġm', 'b', '8', 'Ġg', '10', 'xs', 'b', 'ss', 'Ġrev', 'o', 'Ġ8', 'x', 'sw', 'ss', 'Ġaxis', '12', 'Ġaxis', 'Ġ65', 'Ġrecon', 'Ġ6', 'bg', 'Ġrev', 'o', 'Ġ12', 'Ġhp', 'bs', '4', 'Ġv', '2', 'Ġg', '10', 'x', 'ss']	49
XLNet	['_bra', 'vo', 'x', '_loudspeaker', 's', 'as', '_follow', '_', 'm', 'ali', 'bu', '_', 'mb', '12', '_v', '2', '_', 'm', 'ali', 'bu', '_', 'mb', '8', '_', 'g', '10', 'x', 's', 'bs', 's', '_', 're', 'vo', '_8', 'x', 's', 'ws', 's', '_', 'axis', '12', '_', 'axis', '_65', '_recon', '_6', 'b', 'g', '_', 're', 'vo', '_12', '_', 'h', 'p', 'bs', '4', '_v', '2', '_', 'g', '10', 'x', 's', 's']	65

Table 3.6 Comparison of tokenisation outputs from different Transformer pretrained models. The table shows how a sample product description is tokenized by various models, revealing differences in the form of tokens and unique numbers of tokens produced due to their distinct tokenization algorithms.

Following token encoding, the tokens were converted into token embeddings and

passed through the Transformer encoder block, which consists of sequentially stacked layers that vary in depth depending on the architecture of each pretrained language model. Attention weights were then assigned to the token embeddings via the self-attention mechanism, a core component of Transformer models. The attention patterns of some pretrained language models can be visualised using tools such as BertViz (Vig, 2019), as illustrated in Figure 3.2. The processing of token embeddings continued through several additional components, such as FFNs and layer normalization, aligned with the architectural design to produce hidden states used for downstream tasks. The resulting hidden states, contextual representations of the input, are learned during pre-training on a large general-purpose corpus, and the specific learning objective task of pretrained language models forms this representation. These pretrained language models can then be fine-tuned for task-specific applications.

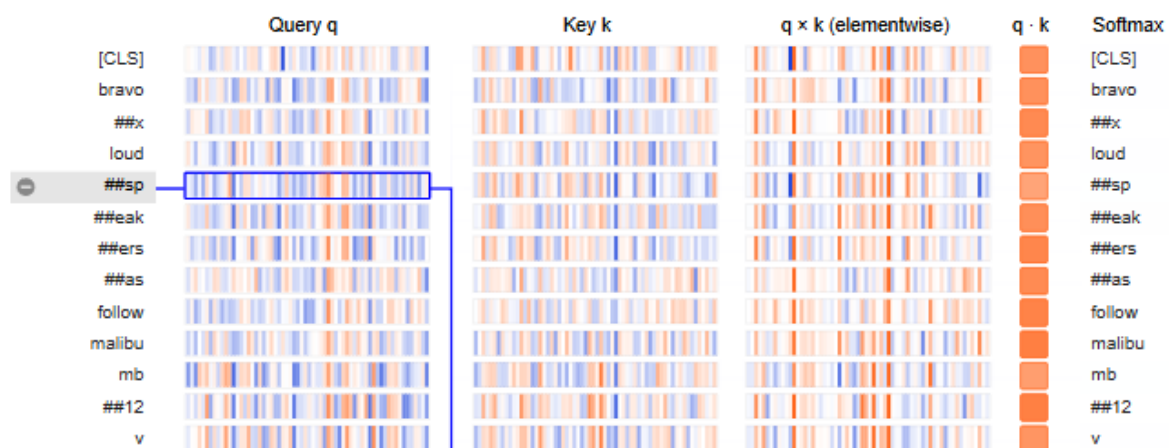


Figure 3.2 Visualisation of the self-attention mechanism in a Transformer model using BertViz. This figure illustrates how the model dynamically allocates attention weights to different tokens, revealing the model's process of attending to other parts of the input sequence during the token encoding process.

Two primary transfer learning approaches can be utilised to adapt pretrained language models for specific downstream tasks, which are divided as follows: feature extraction and full fine-tuning (Qiu *et al.*, 2020; Zhuang *et al.*, 2021). In the feature extraction approach, pretrained models are treated as off-the-shelf feature extractors. It allows the embeddings extracted from pretrained language models to be used directly as input features for separate machine learning algorithms to perform the target task (Mars, 2022). Alternatively, fine-tuning involves adjusting the weights of the pretrained

language model based on task-specific data for the downstream task, which is the approach implemented in this thesis. Fine-tuning pretrained models on domain-specific tasks offers advantages leading to improved model performance and faster convergence (Li, Lei, *et al.*, 2022; Subakti *et al.*, 2022). Besides enhancing contextual representations, it also often requires less labelled data while still achieving strong results (Devlin *et al.*, 2019). Additionally, fine-tuning methods can help mitigate the negative impact of imbalanced datasets (Arslan and Cruz, 2024; Devlin *et al.*, 2019). For all these reasons, this study adopted a fine-tuning approach, which is commonly used in prior work (Devlin *et al.*, 2019; Dodge *et al.*, 2020; Sun *et al.*, 2020).

Input format was crucial for fine-tuning pretrained language models because each model requires a specific input format and batch consistency for efficient computation. Padding tokens [PAD] were added to shorter sequences to ensure uniform sequence lengths across the batch. An attention mask was used to differentiate real tokens from padding tokens to ensure that padding tokens were ignored during attention computation. Separator tokens [SEP] mark the end of sentences or segments. A classification token [CLS] or <s> is added at the beginning of the input sequence. This token provided an aggregate representation of the entire sequence and is typically used as input to the classification head for classification tasks. The properly formatted input was then used to fine-tune the model, where the process involved adjusting the weights of the learned language representations for text classification.

3.2.5 Classification Approaches

The fine-tuned text encoder outputs were processed through a classification head made of fully connected layers. It used non-linear activation functions and a final linear layer to map to the target classes, allowing the model to learn complex decision boundaries for accurate label predictions. Two methods are established in this study: flat and hierarchical classification.

Flat Classification

In the first proposed method, an overview of the flat classification network configuration can be seen in Figure 3.3. The classification head consisted of three fully connected layers with dimensions 768, 1024, and 2048, followed by an output layer of size C , where C denotes the number of possible classes. The [CLS] token

representation h_{cls} was passed through a three-layer classification head during the forward pass. The logits were computed as: $z = W_3 \left(\text{ReLU}(W_2(\text{ReLU}(W_1 h_{cls} + b_1)) + b_2) \right) + b_3$ and the probability distribution over C classes was then obtained using softmax: $\hat{y} = \text{softmax}(z)$, $\hat{y}_i = \frac{\exp(z_i)}{\sum_{j=1}^C \exp(z_j)}$. ReLU activation functions and dropout were incorporated between the linear layers. The training procedure employed the AdamW optimiser (Loshchilov and Hutter, 2017), and the loss function employed categorical cross-entropy loss: $\mathcal{L}_{CE} = -\sum_{i=1}^C y_i \log(\hat{y}_i)$ where y_i is the true label and $\hat{y}_i$ is the predicted probability for class i . The model was trained for a few epochs and typically achieved validation loss convergence (Sun *et al.*, 2019).

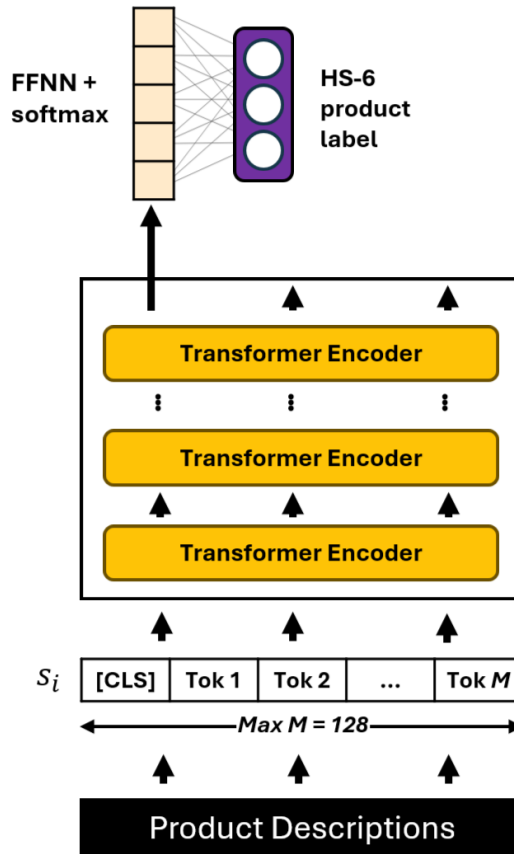


Figure 3.3 Proposed flat classification network for trade product classification. This approach fine-tune pretrained Transformer-based models with a classification head composed of fully connected hidden layers on top of the encoder.

Hierarchical Classification

The second approach, hierarchical classification, was employed to predict each level

within the HS code structure, namely, Chapter, Heading, and Sub-heading. This method leveraged the inherent hierarchical structure of HS codes, where predictions at each level can inform and constrain predictions at subsequent levels, offering the advantage of decomposing a complex multi-class classification task into a series of simpler, more manageable decisions. By progressively predicting broader categories before narrowing down to more specific subcategories, the model was expected to improve both accuracy and interpretability. Additionally, this hierarchical classification approach enhanced scalability, making it suitable for handling a large number of classes and increasing coverage across a wide range of diverse product categories.

We propose a novel method for hierarchical classification based on a local classifier per level strategy (Silla and Freitas, 2011), which involves training a separate multi-class classifier for each level of the class taxonomy. This approach was considered a suitable fit for our problem structure, while being capable of capturing the predefined data taxonomy with a high level of detail (Barbosa, 2021). Three independent classifiers, one for each HS code level, were developed, where the level is denoted as $l \in \{1,2,3\}$ to represent classifiers for Chapter, Heading, and Sub-heading, respectively. The hierarchical classification process initiated by extracting the embeddings from the text input. The [CLS] token embedding, $h_{cls} \in \mathbb{R}^d$, served as the aggregate representation of the input sequence, where d denotes the dimensionality of the embeddings. The embeddings corresponding to the [CLS] token were processed through a level-specific network to generate hierarchy-aware representations: $h^{(l)} = W^{(l)}h_{cls} + b^{(l)}$, where $h^{(l)} \in \mathbb{R}^d$ represents the level-specific transformation for level l . For the level 1 classifier (Chapter), the logits were computed directly from the level-specific representation: $z^{(1)} = W_1^{(1)}h^{(1)} + b_1^{(1)}$. The level 2 classifier (Heading) incorporated information from both levels by concatenating their representations: $h_{concat}^{(2)} = [h^{(1)}; h^{(2)}] \in \mathbb{R}^{2d}$, and the logits were computed as: $z^{(2)} = W_2^{(2)}h_{concat}^{(2)} + b_2^{(2)}$. Similarly, the Level 3 classifier (Sub-heading) concatenated representations from all three levels: $h_{concat}^{(3)} = [h^{(1)}; h^{(2)}; h^{(3)}] \in \mathbb{R}^{3d}$, and the logit for this level was obtained from this process: $z^{(3)} = W_3^{(3)}h_{concat}^{(3)} + b_3^{(3)}$. For each level l , the probability distribution over classes was computed using softmax: $\hat{y}^{(l)} = \text{softmax}(z^{(l)})$.

The model was trained using a combination of standard categorical cross-entropy loss and a customised hierarchical loss. The categorical cross-entropy loss was computed as the sum of losses across all three hierarchical levels: $\mathcal{L}_{CE} = \sum_{l=1}^3 \mathcal{L}_{CE}^{(l)} = \mathcal{L}_{CE}^{(1)} + \mathcal{L}_{CE}^{(2)} + \mathcal{L}_{CE}^{(3)}$. Each level-specific loss was defined as: $\mathcal{L}_{CE}^{(l)} = -\sum_{i=1}^{C^{(l)}} y_i^{(l)} \log(\hat{y}_i^{(l)})$, where $y_i^{(l)}$ is the true label and $\hat{y}_i^{(l)}$ was the predicted probability for class i at hierarchical level l , with $C^{(l)}$, representing the number of classes at level l .

An additional customised hierarchical loss was implemented to penalise the prediction that violates the hierarchical structure, which was defined as: $\mathcal{L}_{Hie} = \mathcal{L}_{Hie}^{(1,2)} + \mathcal{L}_{Hie}^{(2,3)}$. The hierarchical loss consisted of two components: the hierarchical loss between levels 1 and 2: $\mathcal{L}_{Hie}^{(1,2)} = \sum_{i=1}^N (1 - \delta_i^{(1,2)}) \cdot \mathbb{1}[\tilde{y}_i^{(2)} \notin \text{children}(y_i^{(1)})]$, where $\tilde{y}_i^{(2)} = \text{argmax}(\hat{y}_i^{(2)})$ was the predicted Heading for sample i , $\delta_i^{(1,2)}$ was the predicted probability of the chosen Heading and $\text{children}(y_i^{(1)})$, represents the valid Heading under the true label of chapter for sample i . The hierarchical loss between levels 2 and 3: $\mathcal{L}_{Hie}^{(2,3)} = \sum_{i=1}^N (1 - \delta_i^{(2,3)}) \cdot \mathbb{1}[\tilde{y}_i^{(3)} \notin \text{children}(y_i^{(2)})]$, where $\tilde{y}_i^{(3)} = \text{argmax}(\hat{y}_i^{(3)})$ was the predicted Sub-heading for sample i , $\delta_i^{(2,3)}$ was the predicted probability of the chosen Sub-heading and $\text{children}(y_i^{(2)})$, represents the valid Sub-heading under the true label of Heading for sample i .

The hierarchical classification forced the model to learn the structural information of the product category hierarchy, providing the advantage of obtaining complete information for category predictions that progress from broader categories (chapters) to specific 6-digit HS codes (sub-headings). This approach provides traceability in the model's predictions, enabling better interpretation of classification decisions. The hierarchical classification approach is applied only to the selected pretrained language model that achieves the best performance in the flat classification approach.

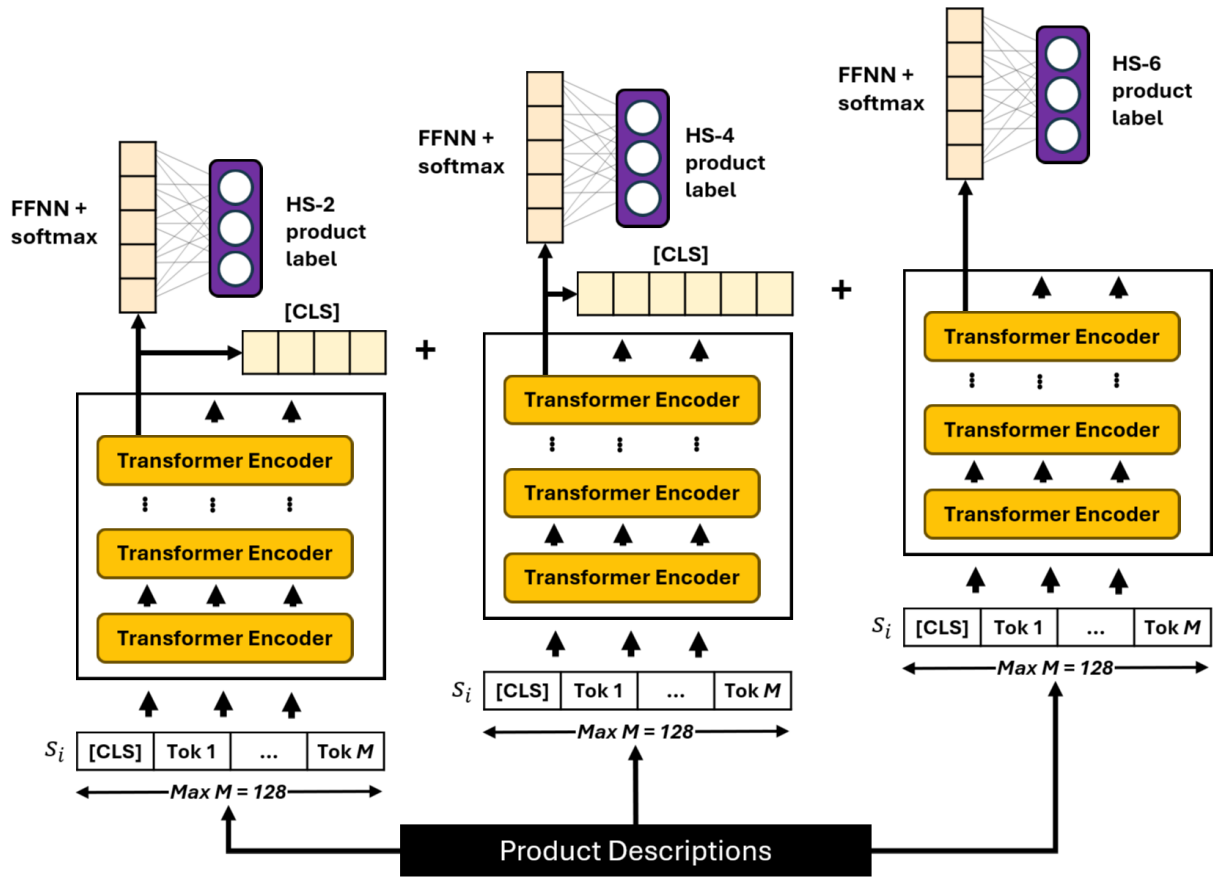


Figure 3.4 Proposed hierarchical network architecture for trade product classification. The classification system consists of three levels (Chapter HS-2, Heading HS-4, Sub-heading HS-6), each employing a transformer-based model as the base encoder. The network processes input and predicts labels at three hierarchical levels, with the output from each level passing to the subsequent level, enabling structured, level-wise classification.

3.2.6 Experimental Setting

Experimental Setup

All experiments and data analysis procedures were conducted on NVIDIA Tesla P100 GPUs with 16 GB of RAM, utilising the computing facilities at Cardiff University, managed by Advanced Research Computing at Cardiff (ARCCA). The pretrained language models were provided and downloaded from Hugging Face¹⁵ (Wolf *et al.*, 2020), a repository hub offering various pretrained models based on the Transformer architecture. The base-uncased models were case-insensitive and were preferably

¹⁵ <https://huggingface.co/>

used in this experiment, where case variations are not semantically significant for product descriptions. The hyperparameters configured for the classification head, including epoch, learning rate, batch size, and maximum sequence length, are presented in Table 3.7.

Hyperparameter	Value	Hyperparameter	Value
Embedding dimension	768	Learning rate	2×10^{-5}
Max sequence length	128	Dropout	0.2
Epoch	4	Optimizer	Adam
Batch size	16	Activation Function	ReLU

Table 3.7 Default hyperparameter configuration used for the trade product classification task.

Baselines

We benchmarked the performance of transformer-based models against traditional methods for the evaluation of text representation purposes. The baseline model used consists of TF-IDF (Salton and Buckley, 1988), Word2vec (Mikolov *et al.*, 2013), GloVe (Pennington *et al.*, 2014), and FastText (Bojanowski *et al.*, 2017). The encoded textual data were fed into a similar feed-forward neural network-based classification head with the same classification configuration used in our proposed approach, and the models were trained with the configurations shown in Table 3.7. For non-transformer-based language models, we used pretrained models trained on English corpora, as shown in Table 3.8, except for TF-IDF, which we built from scratch.

Language Models	Dimension	Source pretrained
TF-IDF	10,000	Applied using Scikit-learn with 10,000 top vocabularies, extracted unigram and bigram features.
Word2vec	300	https://code.google.com/archive/p/word2vec/
GloVe	300	https://nlp.stanford.edu/projects/glove/
FastText	300	https://fasttext.cc/docs/en/english-vectors.html

Table 3.8 Baseline language models used to generate text representations.

Metrics

To evaluate the performance of both flat and hierarchical classification approaches for

HS code prediction, we used precision, recall, and weighted F1-score metrics with weighted-average metrics that account for class imbalance inherent in trade classification data. For each level l in the hierarchy, where $l \in \{1, 2, 3\}$ corresponds to Chapter, Heading, and Subheading levels respectively, performance metrics are computed differently. Flat classification only utilises the Subheading level, $l = 3$, for evaluation, while the hierarchical approach enables performance assessment across all three levels of the HS taxonomy. The accuracy at level l is defined as $\text{Accuracy}^{(l)} = \frac{1}{N} \sum_{j=1}^{C^{(l)}} \mathbb{1}[\hat{y}_j^{(l)} = y_j^{(l)}]$, where N represents the total number of samples, $\hat{y}_j^{(l)}$ denotes the predicted class for sample j , and $y_j^{(l)}$ is the true class label. The weighted precision, recall, and F1-score are computed as: $\text{Precision}^{(l)} = \frac{1}{N} \sum_{i=1}^{C^{(l)}} n_i^{(l)} \cdot \frac{\text{TP}_i^{(l)}}{\text{TP}_i^{(l)} + \text{FP}_i^{(l)}}$, $\text{Recall}^{(l)} = \frac{1}{N} \sum_{i=1}^{C^{(l)}} n_i^{(l)} \cdot \frac{\text{TP}_i^{(l)}}{\text{TP}_i^{(l)} + \text{FN}_i^{(l)}}$, and $\text{F1}^{(l)} = \frac{1}{N} \sum_{i=1}^{C^{(l)}} n_i^{(l)} \cdot \frac{2 \cdot \text{Precision}_i^{(l)} \cdot \text{Recall}_i^{(l)}}{\text{Precision}_i^{(l)} + \text{Recall}_i^{(l)}}$, where $C^{(l)}$ denotes the number of classes at level l , $n_i^{(l)}$ represents the number of samples belonging to class i and $\text{TP}_i^{(l)}$, $\text{FP}_i^{(l)}$, and $\text{FN}_i^{(l)}$, correspond to true positives, false positives, and false negatives for class i , respectively.

To compare flat and hierarchical classification performance, an evaluation followed a 10-fold cross-validation (CV) protocol (Kohavi, 1995), which systematically partitions data into 10 equal subsets, as seen in Figure 3.5. Each fold employed one subset for testing while training on the remaining nine subsets, with this process repeated across all 10 iterations. Performance results are reported as mean $\pm$ standard deviation across folds to quantify both central tendency and variability in model performance.

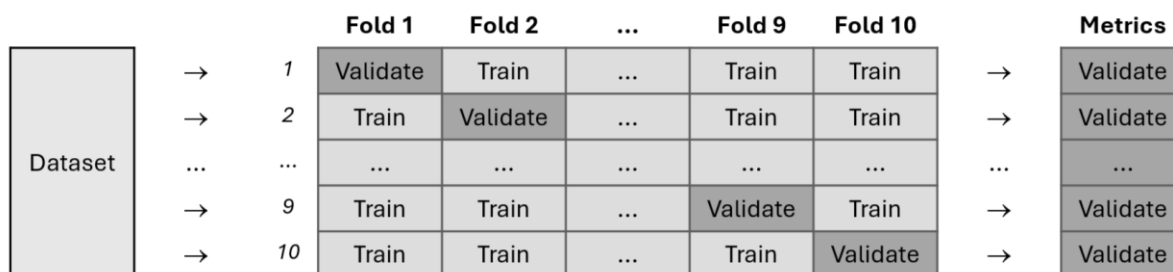


Figure 3.5 10-fold cross-validation process for model evaluation, using rotating train-validation splits to ensure robust performance estimation.

The performance differences were analysed between models at the 5% significance

level using both parametric and non-parametric tests, with the paired Student's t-test (Dietterich, 1998) used for normally distributed performance differences and the Wilcoxon signed-rank test (Demšar, 2006) served as a non-parametric option for non-normal distributions. Additionally, Cohen's d (Cohen, 1988) was calculated to quantify the effect size, facilitating the interpretation of performance improvements.

3.3 Result and Discussion

This section presents the outcomes of the experimental evaluation, composed of an analysis of preprocessing techniques and models classification performance. The following result, the hierarchical classification and flat classification method, reveals their effectiveness in the model performance.

3.3.1 Preprocessing Results

The initial experimental results focus on different preprocessing strategies that affect fine-tuning performance under the flat classification method. The proposed method was compared to traditional text representation. Table 3.9 presents a summary of the findings and the model's performance across preprocessing strategies.

Model	Preprocessing Strategy		
	PP-1	PP-2	PP-3
<i>Baseline:</i>			
TF-IDF	0.627	0.600	0.592
Word2vec	0.477	0.486	0.489
GloVe	0.466	0.470	0.467
FastText	0.478	0.508	0.516
<i>Proposed method:</i>			
ALBERT	0.682	0.630	<u>0.613</u>
BERT	0.736	<u>0.675</u>	<u>0.667</u>
DeBERTa	0.704	0.626	0.619
DistilBERT	<u>0.741</u>	0.671	0.663
ELEKTRA	0.697	0.626	0.616
RoBERTa	0.719	0.643	0.631
XLNET	0.721	0.647	0.636

Table 3.9 Model accuracy results in the classification performance of transformer-based and baseline models across three preprocessing strategies.

All seven transformer-based models outperformed the best traditional baseline. DistilBERT delivered the highest performance across preprocessing strategies, with PP-1 achieving the best accuracy of 74.10%. The PP-2 showed moderate performance decline, and PP-3 obtained the lowest scores across all metrics, with accuracy dropping to 61.34%. Overall, these results suggest that overly aggressive text preprocessing by removing product attributes has a significant impact on the model classification performance, as applied with PP-3. The decline in DistilBERT's performance between PP-1 and PP-3 (from 74.10% to 66.30%) underlined the risks of aggressive data transformation during preprocessing. Text preprocessing that involves product attributes should be removed with caution, as those that appear noisy may still contain valuable signals for the classification model. The performance difference between the best and worst preprocessing methods was up to 12%, emphasising the critical role of effective preprocessing strategies.

In the baseline model analysis, TF-IDF achieved a classification accuracy of 62.69%, which suggests that sparse representation outperformed dense representation utilising trade data. The non-transformer-based models with dense representations performed poorly, with Word2Vec achieving only 47.65% accuracy, and GloVe and FastText showed similar performance. The gap in the model performance indicates that non-contextualised embeddings were difficult to capture the domain-specific terminology for HS code classification.

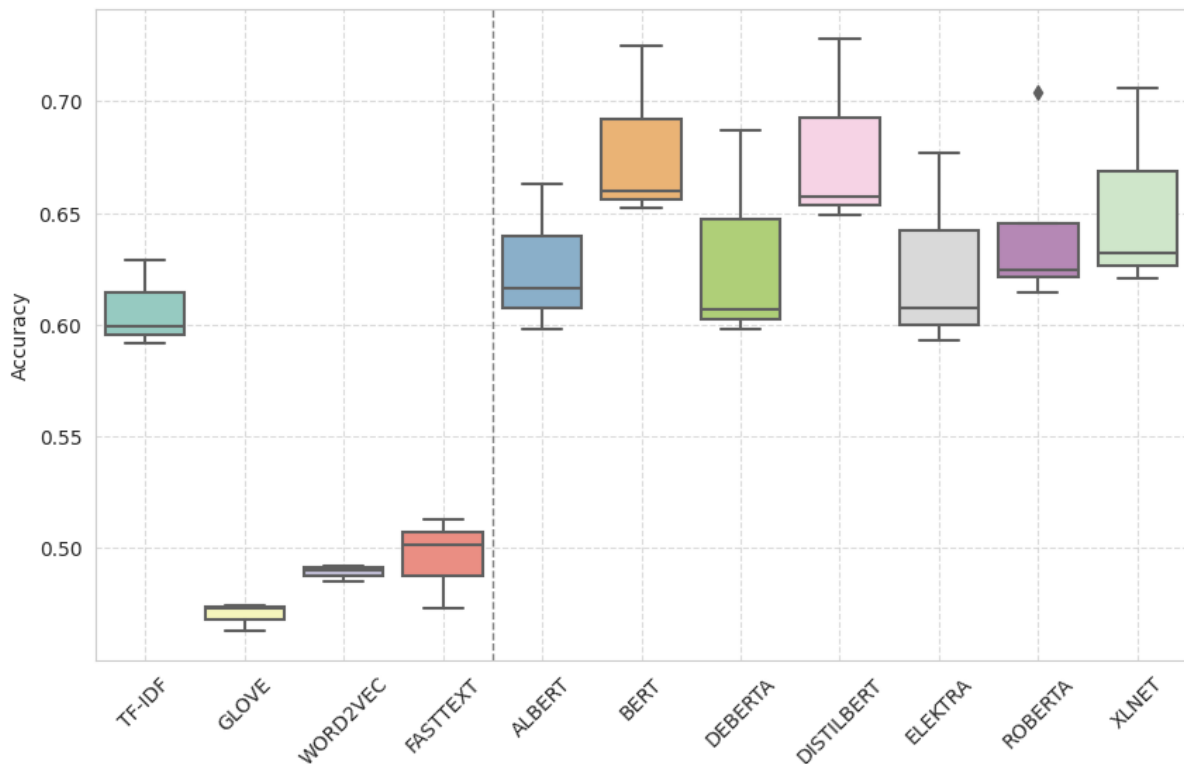


Figure 3.6 Model performance of fine-tuned models on the trade dataset across different preprocessing strategies.

The results also indicate that transformer-based models are more sensitive to preprocessing variations than non-transformer-based models. As shown in Figure 3.6, the boxplot charts visualise the range of impact that different preprocessing strategies have on model performance. Our finding aligns with a recent study that found text preprocessing substantially influences modern language model performance, including transformer-based models (Siino *et al.*, 2024). The results underscore the importance of careful preprocessing design because minor modifications can yield significant performance variations in transformer models using the trade dataset. To further investigated the implications of different preprocessing strategies, we examined the number of tokens generated. Transformer-based models employed different tokenisation algorithms, which explains the differences in the number of tokens generated by each model, as demonstrated in Table 3.10. The average percentage decreased in the number of tokens across all models from PP-1 to PP-2 is 47.27%, underlining a substantial reduction in tokens, primarily due to removing numerical elements from the text. When related to model performance, this treatment corresponds to an average performance drop of 9.67%. This result indicates that

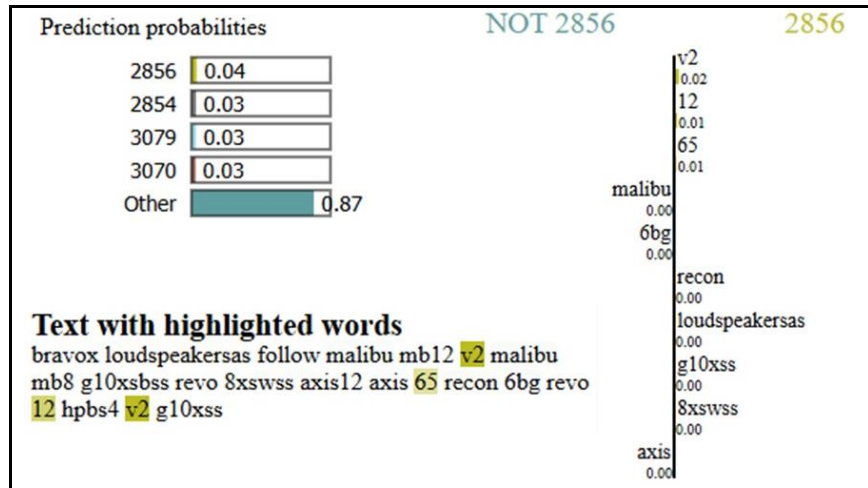
numerical product attributes provide meaningful signals for the model's ability to predict class categories correctly. Such numbers likely correspond to product attributes like the product size, the number of products, or HS code references embedded in the descriptions. Notably, a prior study using a transformer-based model favoured removing numbers, codes, and part numbers from product descriptions during preprocessing for HS code classification (de Lima *et al.*, 2022), which was contrary to our findings to preserve this information for optimal model performance. The preprocessing step from PP-2 to PP-3 focused on eliminating supplementary product attributes, leading to a further average token reduction of 24.07%. This change caused only a 1.53% average drop in classification performance, indicating that the information removed in this stage has a limited influence on the model's ability to determine the correct product category compared to the removal of numerical elements.

Model	Tokenization Algorithm	Number of tokens			Accuracy		
		PP-1	PP-2	PP-3	PP-1	PP-2	PP-3
ALBERT	SentencePiece	25.46	13.81	10.45	0.682	0.630	0.613
BERT	WordPiece	26.04	13.13	10.01	0.736	<u>0.675</u>	<u>0.667</u>
DeBERTa	SentencePiece	22.10	12.05	8.93	0.704	0.626	0.619
DistilBERT	WordPiece	26.04	13.13	10.01	<u>0.741</u>	0.671	0.663
ELEKTRA	WordPiece	26.04	13.13	10.01	0.697	0.626	0.616
RoBERTa	Byte Pair Encoding	24.74	13.53	10.35	0.719	0.643	0.631
XLNET	SentencePiece	28.23	15.36	11.75	0.721	0.647	0.636
Average difference PP1 & PP2		↓ 47.27%			↓ 9.67%		
Average difference PP2 & PP3		↓ 24.07%			↓ 1.53%		

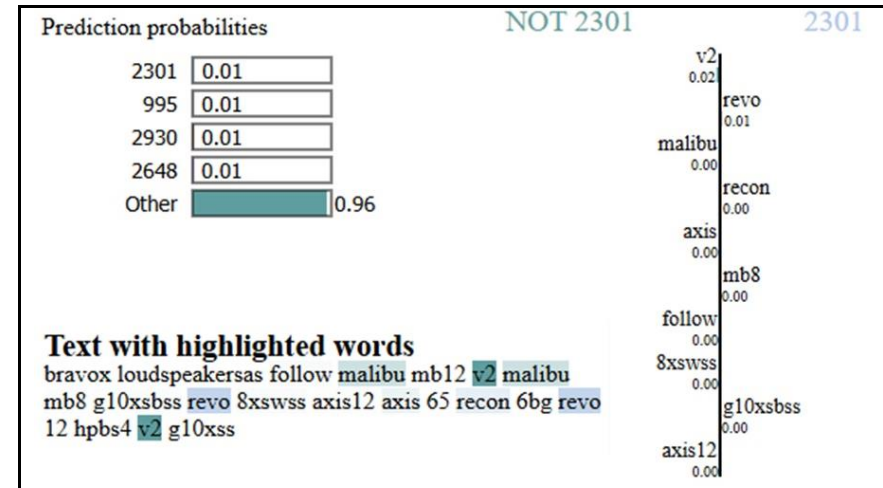
Table 3.10 Number of tokens generated by pretrained language models for the training dataset (in millions) across different preprocessing strategies and their impact on model accuracy.

To enhance model explainability, we employed the LIME explainer (Ribeiro *et al.*, 2016) to analyse the contribution of individual text elements (words or tokens) to final predictions. This approach visualises the model's predictive probabilities for selected classes and explains model behaviour in predicting product codes. A real-world trade transaction sample was used for this analysis with the product description: "*bravox loudspeakers as follow malibu mb12 v2 malibu mb8 g10xsbs revo 8xswss axis12 axis*".

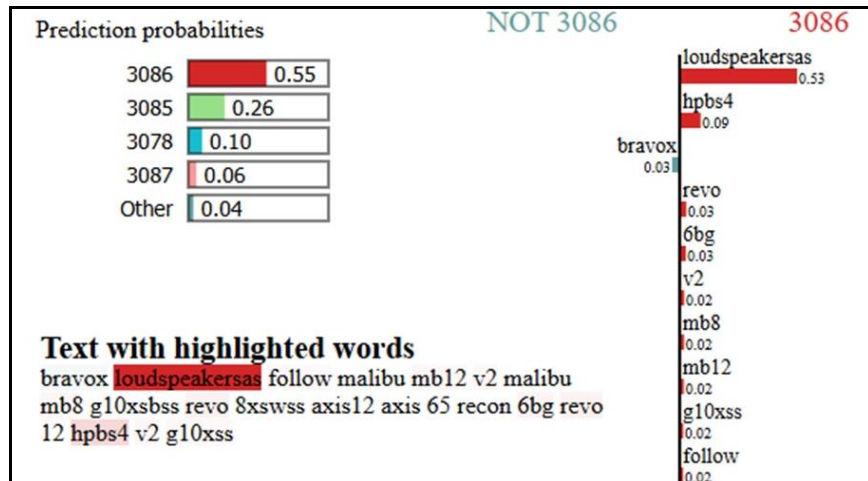
65 recon 6bg revo 12 hpbs4 v2 g10xss", which has a label of 3085 corresponding to HS code 851762 with the explanation in the HS codebook: "*Single loudspeakers, mounted in their enclosures*".



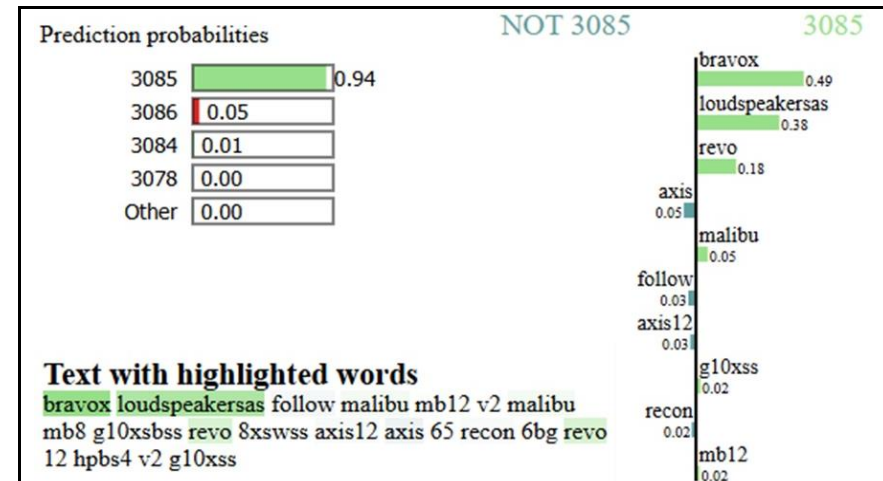
a. LIME explainer result with TF-IDF as encoder



b. LIME explainer result with Word2vec as encoder



c. LIME explainer result with ALBERT as encoder



d. LIME explainer result with DistilBERT as encoder

Figure 3.7 LIME feature visualisation of models using TF-IDF, Word2vec, ALBERT and DistilBERT to display the prediction probabilities assigned to a particular class based on the whole input text.

Traditional methods present evident limitations with these heterogeneous product description attributes. Both the TF-IDF-based and Word2Vec-based classifiers incorrectly predict the product description as belonging to classes 2856 and 2301, respectively, as illustrated in Figure 3.7 (a) and Figure 3.7 (b), indicating limitations in handling noisy or ambiguous input using traditional embedding methods. Using the transformer-based model ALBERT, shown in Figure 3.7 (c), the model assigned contribution weights to certain words that influenced the class prediction, despite resulting in an incorrect prediction. In contrast, DistilBERT, as shown in Figure 3.7 (d), correctly predicted the code with a high probability score, with the most influential word elements, such as "*bravox*" and "*loudspeakersas*". The term "*loudspeakersas*" is likely a misspelling of "*loudspeaker as*". Nevertheless, both ALBERT and DistilBERT were able to identify the product type using their subword tokenization. This finding shows that transformer-based models can handle product descriptions with spelling errors or mistakes.

3.3.2 Transformers-based Models

Models Performance

Table 3.11 presents detailed metrics of the performance for transformer-based models. In terms of weighted average metrics, which account for class imbalance in the dataset, DistilBERT demonstrates superior performance to achieve the highest precision (74.24%), recall (74.10%), and F1-score (73.30%). This result is remarkable since DistilBERT is a distilled version of BERT, achieving superior performance despite having fewer parameters and faster inference speed. The experimental results of the performance of transformer-based models indicate that architectural design plays an important role in capturing key linguistic patterns relevant to the classification task, even when the models were fine-tuned on the same dataset. RoBERTa, BERT, and XLNET show competitive performance, achieving moderate scores close to the top performer. In contrast, DeBERTa, ELECTRA, and ALBERT show relatively lower performance in the weighted average metrics compared to the top performers, despite DeBERTa being known to demonstrate strong performance with consistently high rankings in the General Language Understanding Evaluation (GLUE) dashboard¹

¹ <https://gluebenchmark.com/leaderboard>

across various NLP tasks (Wang *et al.*, 2018). Achieving an optimal classification performance on domain-specific datasets requires a sensible selection of pretrained language models. In this study, DistilBERT and BERT demonstrate the most effective performance for trade product classification, with a 9.6% F1-score difference between the best-performing model (DistilBERT) and the lowest-performing model.

Model	Accuracy	Precision	Recall	F1-score
ALBERT	0.682	0.692	0.682	0.669
BERT	0.736	0.739	0.736	0.727
DeBERTa	0.704	0.704	0.704	0.692
DistilBERT	<u>0.741</u>	<u>0.742</u>	<u>0.741</u>	<u>0.733</u>
ELEKTRA	0.697	0.703	0.697	0.684
RoBERTa	0.719	0.725	0.719	0.709
XLNET	0.721	0.723	0.721	0.711

Table 3.11 Performance of fine-tuned Transformer-based language models across various metrics using preprocessing strategy PP-1 under flat classification.

Table 3.12 compares the performance of different models in predicting HS codes. DistilBERT outperforms all others on 3,328 out of 3,566 HS code categories based on F1-score with the PP-1 preprocessing strategy. BERT ranks second in performance. These results establish DistilBERT and BERT as the top-performing transformer-based models for trade product classification.

Model	ALB	BER	DEB	DIS	ELE	ROB	XLN	Average
ALBERT (ALB)	-	2,454	2,867	2,262	2,931	2,527	2,574	2,603
BERT (BER)	3,463	-	3,424	2,705	3,430	3,166	3,156	3,224
DeBERTa (DEB)	3,189	2,498	-	2,320	3,078	2,656	2,675	2,736
DistilBERT (DIS)	3,491	3,061	3,431	-	3,444	3,259	3,282	3,328
ELEKTRA (ELE)	3,141	2,486	2,987	2,314	-	2,628	2,641	2,700
RoBERTa (ROB)	3,404	2,733	3,277	2,507	3,302	-	2,997	3,037
XLNET (XLN)	3,369	2,738	3,270	2,484	3,305	2,920	-	3,014

Table 3.12 Number of classes in which each model (vertical columns) performs better or equal compared to other models (horizontal columns).

Our proposed models outperformed previous studies using the same dataset. The study by Ruder (Ruder, 2020) is the most comparable to ours, using all available class categories and a feed-forward network classifier, but achieved an F1-score (weighted average) of 61% compared to our best result of 73.30% with DistilBERT, a 12.3% improvement. Another study utilised an identical dataset to ours with much fewer class labels, reporting a lower accuracy of 56% compared to our model performance (Shubham *et al.*, 2023). The other two studies are difficult to compare, despite using a similar dataset, since the former study only utilised HS-4 results, and the latter did not provide a complete evaluation of model performance on the entire dataset (Luppens, 2019; Spichakova and Haav, 2020a).

Training Time

The architectural design of pretrained language models, along with their tokenisation algorithms, results in different parameter sizes, which significantly influence the time required for fine-tuning. This training time is also affected by the dataset size and the token length resulting from the preprocessing strategy. The impact of our preprocessing strategy shows that as preprocessing progresses from PP-1 to PP-3, the number of tokens decreases, leading to shorter training times. Table 3.13 summarises the fine-tuning times for each pretrained model across the three preprocessing strategies. Markedly, DistilBERT achieved the fastest training time across all preprocessing strategies, although ALBERT had the smallest number of parameters among the evaluated language models. In contrast, XLNet recorded the longest training times, even though DeBERTa has the largest parameter size. These findings underline that various pretrained language models process text input differently when fine-tuned on a similar dataset, and the large number of parameters in the language models does not necessarily correlate with the length of training times.

Model	Preprocessing			Parameters Size
	PP-1	PP-1	PP-1	
ALBERT	5:34:05	3:26:47	2:53:18	~12 M
BERT	9:21:20	5:17:05	2:46:32	~110 M
DeBERTa	11:08:33	7:29:37	3:54:25	~139 M
DistilBERT	<u>4:49:42</u>	<u>2:47:48</u>	<u>2:27:56</u>	~66 M
ELEKTRA	5:28:39	5:19:52	2:47:08	~110 M
RoBERTa	8:47:19	5:23:43	4:44:08	~125 M
XLNET	12:35:04	7:26:18	3:57:26	~110 M

Table 3.13 Duration of training time (hours: minutes: seconds) and parameter sizes of pretrained language models. The results reflect how architectural differences and model size influence fine-tuning efficiency when processing datasets with different preprocessing strategies (trained on an NVIDIA Tesla P100 GPU).

Model Performance and Training Time

The relationship between model performance and training time under different preprocessing strategies indicates that combining a pretrained language model and a preprocessing strategy is important to achieve an optimal balance between efficiency and accuracy, as shown in Figure 3.8. The diagram also reveals that a certain preprocessing strategy can decrease training time, but it can also negatively impact the model's performance. DistilBERT proves to be the most favourable model to balance between performance and efficiency among the models, maintaining high classification accuracy while requiring minimal computational resources. The efficiency of DistilBERT is particularly relevant given the scale of trade transaction datasets and the need for repeated model training and evaluation across multiple experimental settings. Moreover, empirical results in this thesis indicate that larger transformer models did not yield marginal performance improvements and resulted in additional computational cost. From a deployment standpoint, the reduced model size and lower resource requirements of DistilBERT make it more suitable for integration into operational customs and trade administration environments, where hardware availability and infrastructure capacity are often constrained. Consequently, DistilBERT is the best choice among the transformer-based models due to its ability to maintain competitive performance while reducing computational complexity.

These findings establish foundations in developing efficient text classification models when utilising domain-specific data, such as trade data, which could be applicable to replicate in other domains that have similar characteristics, such as e-commerce or the medical domain. The best language model performer served as a baseline for comparison with the hierarchical classification approach examined in the following section.

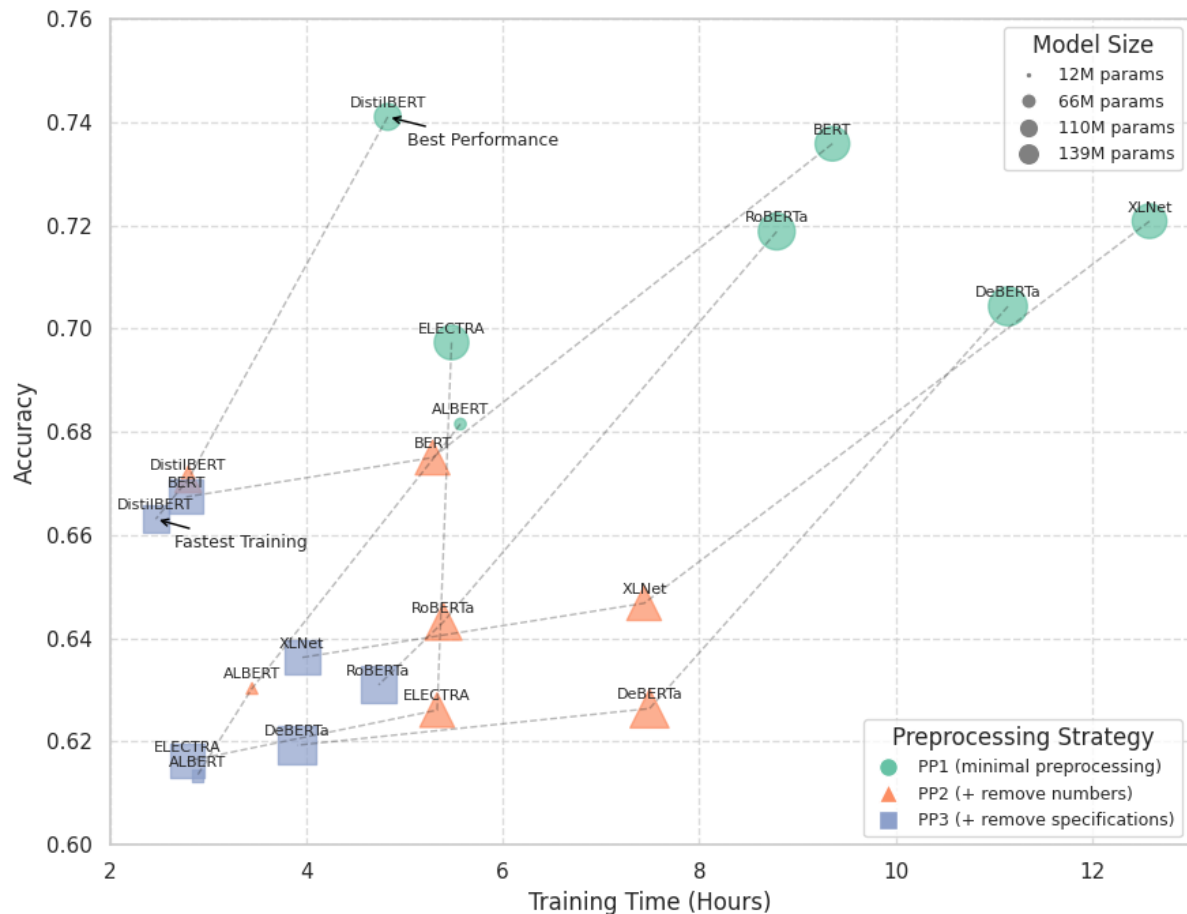


Figure 3.8 Comparison of model performance versus training time using various transformer-based models with different preprocessing strategies with trade data. Fine-tuned DistilBERT model achieved both high performance and fast training time relative to other approaches.

3.3.3 Hierarchical Classification

This section presents the result of the implementation of hierarchical classification for product classification, where DistilBERT was represented as the language model for this evaluation based on the model's effectiveness and efficiency in the flat classification setting. Unlike the flat classification approach that evaluates performance at the HS-6 level, the hierarchical classification approach provides

performance metrics across all three levels of the HS code taxonomy: Chapter (HS-2), Heading (HS-4), and Sub-heading (HS-6).

The performance of the proposed method employing hierarchical flat classification is shown in Table 3.14. At the HS-2 level, representing the broadest classification tier, the model achieved an accuracy of 89.91% and a similar value of F1-scores. This high-performance result was consistent with our expectations, given that HS-2 contains only 89 product categories, making it the fewest classes and the least complex among the three hierarchical levels. The performance at the intermediate level at HS-4 declined slightly, with an accuracy of 84.36% and an F1-score of 84.28%. The performance decline reflects the increased difficulty of the task, as the number of categories expands from 89 at HS-2 to 676 at this intermediate level. At the most detailed tier (HS-6 level), the model achieves an accuracy of 75.57% and an F1-score of 74.95%. This decrease reflects the increased difficulty of the task due to the expanded number of classes (3,566 categories), which is the same number of categories used in the flat classification approach presented earlier.

Model	Class Number	Accuracy	Precision	Recall	F1-score
Chapter (HS-2)	89	0.899	0.899	0.899	0.899
Heading (HS-4)	676	0.843	0.845	0.844	0.842
Sub-heading (HS-6)	3,566	0.755	0.756	0.756	0.749

Table 3.14 Model performance of fine-tuned DistilBERT using the proposed hierarchical classification method. The results highlight model performance at different HS code levels, with an increasing number of classes from Chapter (HS-2) to Sub-heading (HS-6).

3.3.4 Flat and Hierarchical Classification

Evaluation to measure the model's performance both for flat and hierarchical classification was performed using 10-fold CV and the PP-1 strategy. Hierarchical classification outperformed flat classification at the HS-6 level, with statistically significant improvements and large effect sizes for F1-score and accuracy, as seen in Table 3.15.

Model	Flat	Hierarchical	p-value	Mean Difference	Effect Size (Cohen's d)
Accuracy	0.751 ± 0.001	0.763 ± 0.001	< 0.001	0.01	12.99
Precision	0.748 ± 0.002	0.801 ± 0.001	< 0.001	0.05	57.87
Recall	0.751 ± 0.001	0.748 ± 0.001	< 0.001	0.00	-3.71
F1-score	0.742 ± 0.002	0.768 ± 0.001	< 0.001	0.03	29.27

Table 3.15 Model performance using the flat and hierarchical approach for HS-6 codes.

Previous studies have shown inconsistent results about the effectiveness of flat and hierarchical classification approaches. Our findings support this study (Shubham *et al.*, 2023), which utilised hierarchical classification to improve model performance. However, their study reported 56% accuracy for the flat model and 65% for the hierarchical model; this score was lower than what we obtained in this study. While some studies preferred for using flat classification (Ding *et al.*, 2002; Krishnan and Amarthaluri, 2019), other studies achieved better outcomes using hierarchical methods across different datasets related to product data in various domains (Brinkmann and Bizer, 2021; Gao, 2020). A related study applied a transformer-based model to classify product titles of Turkish e-commerce datasets (Ozyegen *et al.*, 2022), but their approach was limited to hierarchical classification only. In addition, Cotacallapa (Cotacallapa *et al.*, 2024) compared flat, hierarchical, and combined classification strategies using traditional machine learning techniques. Markedly, the study recommended the use of transformer-based models, a recommendation that has been addressed in this thesis.

3.4 Summary

Automating product classification is important for enhancing international trade facilitation, particularly in facilitating regulatory compliance through accurate trade code classification. This thesis proposes a machine learning pipeline composed of several key components for trade product classification: the domain-specific preprocessing techniques, the fine-tuned transformer-based language models, and both the flat and hierarchical classification strategies. The results present that text preprocessing is important for text classification, but excessive information removal from product descriptions can reduce transformer-based model performance. This thesis shows that both BERT and DistilBERT achieved competitive performance in

classification tasks. However, the fine-tuned DistilBERT model offers practical benefits through faster training times and efficient computation, providing an optimal balance between classification accuracy and practical effectiveness. Additionally, the novel hierarchical classification approach yields superior performance in terms of accuracy and granularity by generating predictions at various levels of the classification hierarchy. The combined methodological contributions presented in this thesis make them viable candidates for practical implementation in real-world trade classification applications.

The following chapter introduces the use of sentence embeddings to represent trade product descriptions. This approach is expected to improve classification performance by capturing high-quality semantic information compared to word-level embeddings to improve product classification performance.

Chapter 4

Enhancement of Trade Product Classification through Sentence Embeddings and Supervised Contrastive Learning

Following the examination of text classification utilising word-level embeddings in Chapter 3, this chapter addresses the second objective in this thesis by proposing an approach for enhancing sentence-level embeddings to improve trade product classification performance. The proposed method utilised supervised contrastive learning involving a Siamese network architecture to generate sentence embeddings that represent product descriptions. The enhanced sentence embeddings then serve as input features for traditional machine learning algorithms in trade product classification.

4.1 Introduction

Effective NLP-based tasks benefit from the extraction of meaningful and high-quality features from text data, as this impacts overall model performance (Bengio *et al.*, 2013; Szymański, 2014). Among various text representation approaches, sentence embeddings have evolved as an effective method for improving downstream NLP task performance (Conneau and Kiela, 2018), which have been implemented in previous studies across various domains, including information retrieval, question answering, and text classification (Huang *et al.*, 2023; Liu *et al.*, 2023; Mokoatle *et al.*, 2023; Ramesh Kashyap *et al.*, 2024).

A traditional method for creating sentence-level embeddings is to aggregate word-

level representations from methods like Word2Vec (Mikolov *et al.*, 2013), GloVe (Pennington *et al.*, 2014), and FastText (Bojanowski *et al.*, 2017). This technique processes individual word embeddings by the averaging technique, which has been widely used as a straightforward approach offering computational efficiency (Wieting *et al.*, 2016). This method has been used in past studies to classify trade products (Luppens, 2019; Ruder, 2020; Ryzhova and Sochenkov, 2019). However, it has limitations due to the inability to preserve sequential word order and diminish sentence-level semantic content through aggregation operations (Wieting *et al.*, 2016). An alternative method has been developed for generating sentence-level embeddings, such as Doc2Vec (Le and Mikolov, 2014), which is an extension of the Word2Vec framework, producing fixed-length vector representations for variable-length text sequences. This method has been applied in a product classification study (Spichakova and Haav, 2020a). However, this method struggles to capture detailed semantic relationships and contextual dependencies in sentences when compared to more advanced transformer-based models.

Contextualised embeddings generated by transformer-based models, such as BERT, utilise special tokens to represent entire input sequences. BERT introduces a special token, [CLS], at the beginning of each input sequence, and the final hidden representation of this token serves as an aggregate representation of the entire sequence. The use of aggregated tokens to represent a product description sequence has been utilised for trade product classification (de Lima *et al.*, 2022). The results showed a strong model performance for downstream classification tasks. Some studies show that the previous approach does not effectively capture semantic richness and struggles with tasks that require detailed semantic similarity (Ha *et al.*, 2021; Wang and Kuo, 2020).

To improve the capability of sentence embeddings, some researchers have developed context-aware models that involve transformer-based architecture, such as the Universal Sentence Encoder (USE) (Cer *et al.*, 2018) and Sentence BERT (SBERT) (Reimers and Gurevych, 2019). Both methods have been applied in trade classification studies, employing SBERT (Pain, 2021) and USE (Chen, Ben van Rijnsoever, *et al.*, 2021) to generate sentence-level embeddings from product descriptions. However, those studies utilised off-the-shelf models without domain-

specific fine-tuning, which potentially limits their effectiveness in achieving optimal performance when dealing with specialised trade contexts.

Transformer-based models generate embeddings that offer improvements across various downstream NLP tasks, which is aligned with our findings from Chapter 3. However, an issue exists related to these generated embeddings, presenting a phenomenon where embeddings cluster in a narrow region of the vector space, which is referred to as anisotropy. The geometric property of these vector representations is unevenly distributed rather than uniformly spread across the space. In the context of trade product representation, anisotropy is problematic because it reduces the discriminative capacity of embeddings. Product descriptions that are semantically different may appear artificially similar because their vectors are clustered too closely together. This effect is disadvantageous for fine-grained product classification tasks such as HS code assignment, where small semantic distinctions can correspond to different product categories. This drawback reduces the discriminative power of classification tasks to distinguish subtle embedding (Ethayarajh, 2019; Godey *et al.*, 2024; Li, Zhou, *et al.*, 2020). The problem is especially evident in models lacking domain-specific fine-tuning (Rajaei and Pilehvar, 2021). Previous studies have proposed methods to promote uniform distribution of embeddings across the vector space referred to as isotropy (Xiao *et al.*, 2023). The goal is to make embeddings more discriminative leading to improve downstream task performance. An approach to resolving this issue is presented by Gao (Gao *et al.*, 2021), who utilised dropout-based augmentation during training to achieve better quality of sentence embeddings. The other methods proposed a method to transform the embeddings to normalise to follow a standard Gaussian distribution in BERT-flow (Li, Zhou, *et al.*, 2020), and a post-processing technique to remap the embeddings without retraining the model in BERT-whitening (Su *et al.*, 2021). Utilisation of contrastive learning has been introduced in previous studies and advanced the field of sentence representation (Liao, 2021; Moukafih *et al.*, 2022; Zhang, Li, *et al.*, 2021), nevertheless, such methods are constrained by their dependence on negative samples.

This thesis aims to enhance the quality of sentence embeddings by employing contrastive learning. The proposed approach extends existing methods by utilising anchor and positive samples as a pair of samples subsequently used to train the

model. The motivation for this design is grounded in the assumption that positive samples alone provide a sufficient signal for constructing meaningful representations, enabling the model to focus on identifying features that make similar items alike. Furthermore, the inclusion of negative samples presents a challenge in the context of trade datasets, where the large number of classes complicates negative sampling, avoiding unnecessary curated negative samples. The main hypothesis suggests that such an approach constitutes a viable strategy for advancing the effectiveness of semantic representations in downstream classification tasks. The supervised contrastive learning directly optimises the text representation by pulling embeddings of similar instances closer together while pushing dissimilar ones apart. The proposed methodology is anticipated to produce more discriminative sentence-level embeddings in comparison to conventional aggregation approaches, while simultaneously addressing the anisotropy issue. Another advantage of the proposed approach is its flexibility in utilising learned sentence representations for classification tasks. With the proposed method, the learned sentence representations can be further utilised for related downstream tasks, including information retrieval and semantic clustering, without requiring modifications to the underlying architecture or training design.

The principal contributions in this chapter are:

1. This thesis proposes a novel method to enhance sentence embeddings that employs a supervised contrastive learning framework by integrating pretrained transformer-based models involving a Siamese network architecture and several components. The main components include the construction of paired training samples, the deployment of a domain-specific loss function, and the selection of pooling methods.
2. The implementation of the proposed method contributes to enhancing embedding quality, which addresses the anisotropy issue commonly associated with transformer-based embeddings, as indicated by the generated sentence embeddings, which achieved an anisotropy level that is more isotropic than before.
3. Empirical contributions using the proposed sentence embeddings show that the method outperforms well-established fine-tuning methods in terms of classification

accuracy. The method integrates enhanced sentence embeddings with traditional machine learning classifiers, demonstrating the adaptability of the generated embeddings and the efficiency in reducing model training time.

This chapter is outlined as follows. Section 4.2 provides an overview of the proposed model architecture, describing the mechanisms of the key components used in the network. Section 4.3 explains the dataset used in the experiments, model configuration parameters, baseline models, and evaluation metrics. Section 4.4 presents the experimental results, which are divided into three categories: evaluation of embedding quality, model classification performance, and analysis of performance gains attributable to individual model components. Section 4.5 summarises the overall approach to conclude this chapter.

4.2 Methodology

This section describes the proposed methodology, which consists of several components, such as the arrangement of training samples, the configuration of a Siamese network architecture, the application of pooling mechanisms, and the implementation of a task-specific loss function, all of which are interconnected, as illustrated in Figure 4.1. The process to generate sentence embeddings was initiated by arranging pairwise training samples that are subsequently used to train the model, which involves a Siamese architecture utilised for representation learning. The architecture consists of multiple primary components, including the backbone text encoder to generate text features. The encoder is followed by pooling layers, which contribute to generating fixed-sized representations, and the implementation of the objective function used to optimise the model to align similar products closely in embedding space during training. During inference, the fine-tuned model generates enhanced sentence embeddings used for downstream classification tasks, which were evaluated using publicly available trade datasets to validate the effectiveness and generalisation capability of our proposed method.

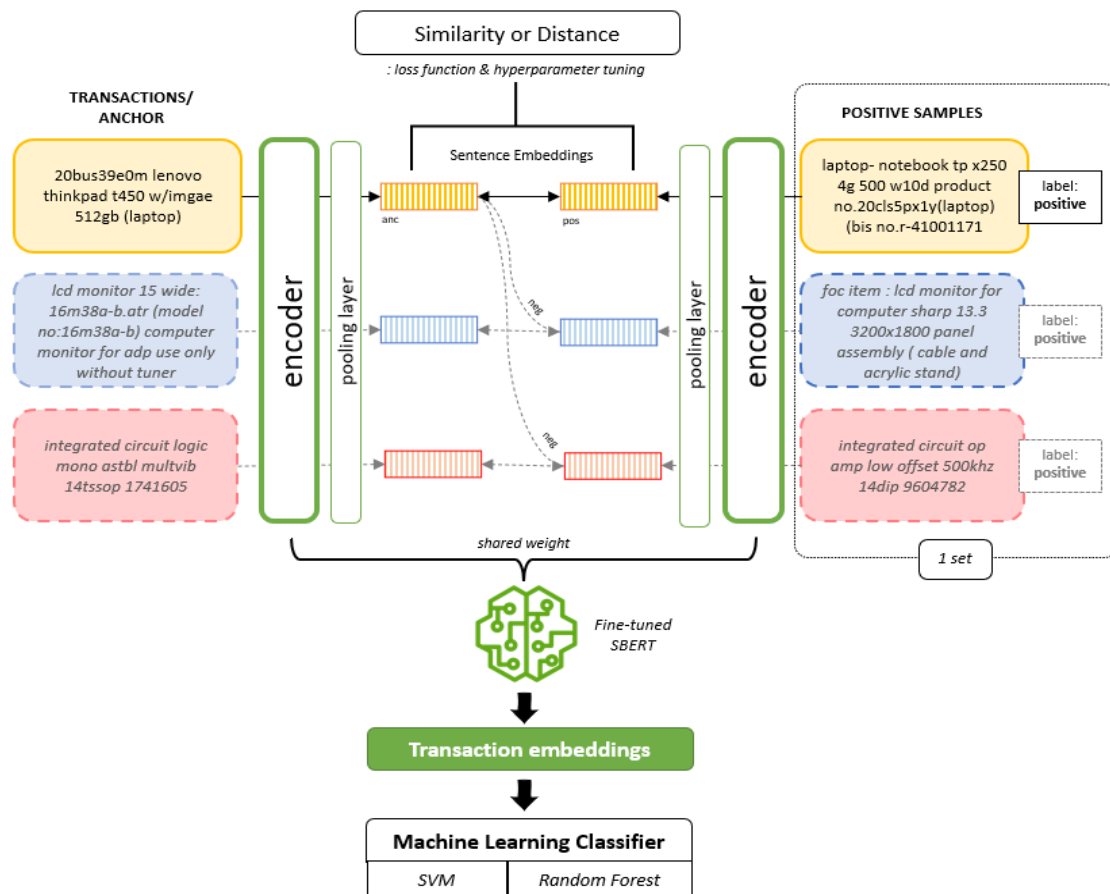


Figure 4.1 Configuration of the proposed method using supervised contrastive learning to generate sentence embeddings for the downstream classification task.

4.2.1 Text Encoder

Our experimental design employed transformer-based language models as the backbone to generate contextualised embeddings from raw text, which offer benefits in capturing the underlying semantics of the data. The choice of backbone encoder is essential to determining the quality of the resulting sentence representations. Accordingly, our text encoder selection was guided by the findings presented in Chapter 3 regarding the most effective language model for trade datasets.

4.2.2 Pooling Layers

Pooling layers are positioned after text encoders, aiming to transform variable-length output embeddings into fixed-size embeddings, as illustrated in Figure 4.1. This pooling mechanism creates sentence-level embeddings by aggregating token-level representations, or optionally by using a projection layer. The transformation of word-

level embeddings into fixed-size sentence representations requires consideration of pooling strategies because the chosen method can impact model performance. Prior research studies demonstrated that mean pooling outperformed max pooling in regression tasks when using transformer-based architectures (Reimers and Gurevych, 2019). On the other hand, empirical evidence showed that max pooling, which selects the maximum value across each dimension, can be advantageous for semantic similarity tasks using BiLSTM-based sentence encoding models (Conneau *et al.*, 2017). Another sentence-level representation approach used a special token, [CLS], that is widely adopted by transformer-based models. Various pooling strategies were implemented in these experiments, including max pooling, mean pooling, and utilising special token representation to examine which pooling methods were advantageous for improving classification performance.

4.2.3 Contrastive Learning

Contrastive learning functions as a framework for generating sentence-level embeddings to optimise similarity between related samples and maximise divergence between dissimilar ones. Our proposed method utilised a Siamese network or bi-encoder architecture, which consists of two identical neural networks that share the same weights, facilitating a contrastive learning implementation (Bhattacharjee *et al.*, 2022; Xu, Xie, *et al.*, 2023). In our implementation, two sentences were fed into the shared network separately, as shown in Figure 4.1, and the framework used was adapted from SBERT (Reimers and Gurevych, 2019), a commonly used framework to perform some tasks, such as semantic similarity and semantic search.

The application of this learning strategy in the trade domain aims to represent the trade product descriptions within a similar class category cluster together in the embedding space. In turn, this method results in embeddings that encourage logical grouping based on semantic similarities at the sentence level. Following this method will enhance the discriminability of the embeddings among different product categories, which is beneficial for utilising them for the downstream task, such as text classification.

4.2.4 Samples Configuration

The implementation of contrastive learning often leverages data in a pairwise or triplet

configuration, and the arrangement of input data samples is an essential factor in sentence representation learning, as it influences the effectiveness of model training (Ramesh Kashyap *et al.*, 2024). Pairs in contrastive learning can be generated through various strategies, including data augmentation (Thakur, Reimers, Daxenberger, *et al.*, 2021). In the domain of vision computing, data augmentation techniques using images can be simple transformations such as rotation, cropping, or colour distortion that preserve semantic meaning. However, when applied to text data type, data augmentation poses inherent challenges due to the discrete nature of such data (Gao *et al.*, 2021). Nevertheless, several methods have been conducted for text augmentation, such as randomly manipulating word positions, as demonstrated in the BART study (Lewis *et al.*, 2020), or an approach that involving back-translation techniques for sentence-level augmentation (Fang *et al.*, 2020), and word-level data augmentation through deleting, reordering, and substituting words in text sequences (Wu *et al.*, 2020). In a recent study, ChatGPT was used to generate sample data for sentence-level data augmentation in few-shot text classification (Dai *et al.*, 2023).

A comprehensive understanding is necessary when selecting to perform data augmentation since they have consequences, particularly for word-level augmentation techniques, which can change the intended meaning of a sentence (Schlegel *et al.*, 2021). In dealing with trade data, it is complicated to perform word-level data augmentation because of the heterogeneous product attributes in product descriptions. The drawback of this approach was also confirmed by our findings in Chapter 3, regarding the sensitivity of domain-specific text for preprocessing and its impact on the model performance. Given these consequences, sentence-level is preferable; however, we opt to use information from the class label to construct pairwise samples, instead of employing dedicated data augmentation techniques.

This thesis proposes to construct pairwise samples using a supervised methodology, where selected samples were guided by class labels in the data. The sample configuration was inspired by a study implemented for binary classification tasks for sentiment analysis (Moukafih *et al.*, 2022). In contrast, our study addressed a classification problem that involved multiple categories spanning hundreds of classes, that were considered a multiclass classification task. In our approach, a pairwise input comprises two samples, referred to as an anchor sample and a positive sample,

originating from transactions in the trade declarations. An anchor sample refers to any transaction in the dataset, and a positive sample is a randomly selected transaction that shares the same class label as the anchor sample. To illustrate our methodology, Figure 4.2 provides a visual representation of pairwise samples construction used for training the model. In our approach, each anchor sample was configured to have one positive sample in the training datasets, but it can be extended to allow an anchor sample to be associated with multiple positive samples, aiming to learn from diverse representations within the same class.

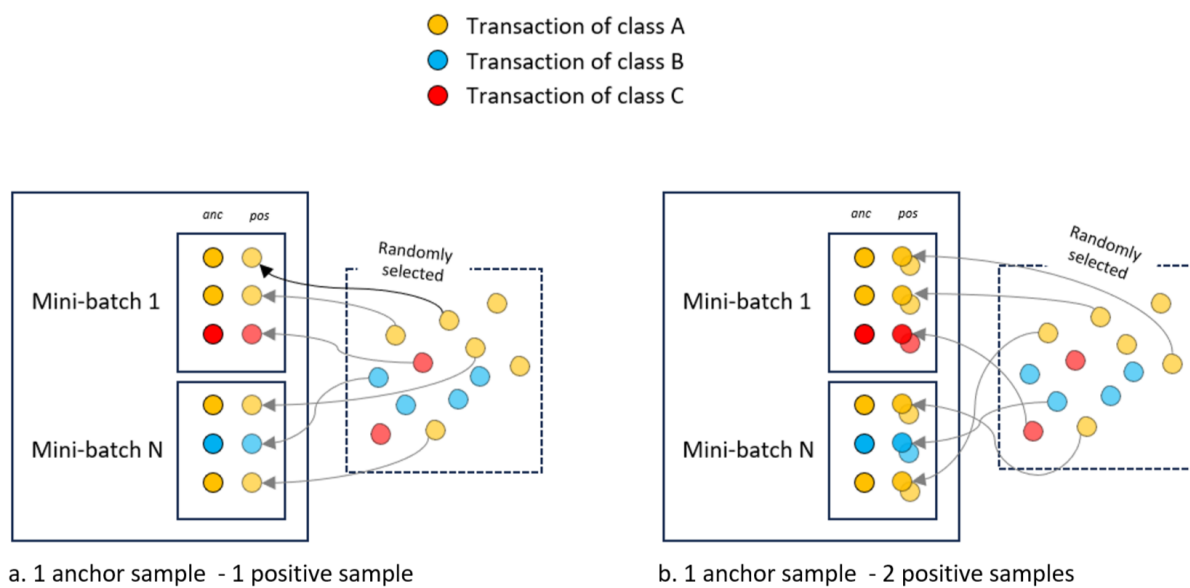


Figure 4.2 The pairwise sample construction process. Each anchor sample is pairing with positive samples, which are randomly selected from the dataset that share the same category (panel a), with the option of multiple positive samples allowed per anchor (panel b) to enable to learn from diverse product descriptions.

4.2.5 Multiple Negative Ranking Loss

The selection of a loss function for model optimisation represents an important factor influencing the model performance (Wang *et al.*, 2022), and the choice of loss functions should be justified on the specific characteristics of the data and task requirements. In multiclass classification, a combination of the softmax activation function and cross-entropy loss has been a standard approach widely adopted in deep learning architectures because of its solid theoretical foundation and proven effectiveness in enhancing a model's ability to predict class labels (Andreieva and Shvai, 2021). Nevertheless, cross-entropy loss has certain limitations, such as its

vulnerability to noisy labels, which consequently impact the model's generalisation performance (Khosla *et al.*, 2021; Pang *et al.*, 2020), and this approach does not explicitly guide the model towards enhancing the quality of its embeddings to improve class separability. In contrast, the primary objective of the loss function used for contrastive learning is to generate semantically enriched embeddings. This is achieved by reducing the distance between embeddings of similar instances and increasing the separation between those of dissimilar instances.

Distance-based loss functions work for representation learning and form the foundation of contrastive learning frameworks. They directly optimise the geometric relationships between data points in the learned embedding space by quantifying similarities and differences between embedded representations (Weng, 2021). The objective is to minimise distances between similar items and maximise distances between dissimilar ones, with some examples including Triplet loss, InfoNCE loss (Oord *et al.*, 2019), and Multiple Negative Ranking (MNR) loss (Henderson *et al.*, 2017).

The MNR loss function operates on batches containing n sentence pairs that are denoted as $[(a_1, p_1^+), (a_2, p_2^+), \dots, (a_n, p_n^+)]$. Each pair (a_i, p_i^+) represents a semantically related positive pair, where a_i serves as the anchor sample and p_i^+ denotes as the corresponding positive sample. Both samples are extracted from a dataset, $\mathcal{D}_{train} = \{(x_j, y_j)\}_{j=1}^N$, where $x_j \in \mathbb{R}^d$, are embeddings and $y_j \in \{1, 2, \dots, C\}$ are class labels. The anchor and positive sample belong to the same category, satisfying the condition $y_{a_i} = y_{p_i^+}$. The loss function aims to minimise the distance between similar pairs, (a_i, p_i^+) while simultaneously maximising the distance between dissimilar pairs (a_i, p_j^+) , where $i \neq j$, which serve as negatives or refer to multiple negatives in a batch-based nature. The loss function is achieved through the formulation:

$$Loss_{MNR} = -\frac{1}{n} \sum_{i=1}^n \log \frac{\exp(\text{scale} \cdot \text{sim}(a_i, p_i^+))}{\sum_{j=1}^n \exp(\text{scale} \cdot \text{sim}(a_i, p_j^+))}$$

The denominator normalises over all positive samples in the batch, where other positive samples from different anchor-positive pairs serve as implicit negative examples. To measure similarity between the vector embeddings, $\text{sim}(\dots)$, represents

the similarity function, where cosine similarity is typically employed for text embeddings due to its scale-invariant properties (Thakur, Reimers, Rücklé, *et al.*, 2021). Since utilising cosine similarity results in values that tend to be relatively small, employing a scaling factor or temperature is one of the options to sharpen the distinctions (Radford *et al.*, 2021). Various scaling strategies have been proposed. Some approaches initialise the temperature at one and gradually increase it during training (Henderson *et al.*, 2020), while SBERT advocates using a fixed scale value of 20 (Reimers and Gurevych, 2019). However, the optimal temperature value depends on the specific task and dataset, typically determined through empirical validation. Algorithm 4.1 presents the key steps in implementing the overall design network in Figure 4.1 during the training phase to generate sentence embeddings.

Algorithm 4.1: Pseudocode of model training with MNR loss

```
1: Input: Datasets of trade transactions consist of batched pairwise anchor and positive samples
2: Output: Trained Siamese network
3: Procedure:
4: Begin
5: For epoch = 1 to epochs do
6:   For batch in batches do
7:     # Encoding phase
8:     anchor_embeddings = base_encoder(anchor_samples)
9:     positive_embeddings = base_encoder(positive_samples)
10:    # Generate pooled vector
11:    pooled_anchor = pooling_function(anchor_embeddings)
12:    pooled_positive = pooling_function(positive_embeddings)
13:    # Compute all-pairs similarity matrix [batch_size, batch_size]
14:    similarity_matrix = similarity_function(pooled_anchor, pooled_positive)
15:    # Compute MNR loss:
16:    # In using MNR loss, diagonal elements of similarity_matrix are true pairs:
17:    # similarity_matrix [ $a_i$ ,  $p_i^+$ ]  $\rightarrow$  positive pair
18:    # similarity_matrix [ $a_i$ ,  $p_j^+$ ] where  $i \neq j \rightarrow$  negative pairs (in-batch negatives)
19:    scores = similarity_matrix * scale
20:    # Each anchor_i should match with positive_i (diagonal elements)
21:    diagonal_labels = range(batch_size)
22:    # MNR loss is minimised when, for each anchor, maximise its similarity with the correct positive, while minimising similarity with all other positives (treated as negatives)
24:    Loss = CrossEntropyLoss(scores, diagonal_labels)
25:    # Calculate gradients and update the network
26:  End
27: End
28: End
```

4.3 Experiments Settings

4.3.1 Datasets

Two publicly available datasets were utilised for this study, obtained from different sources to support the generalisability of our proposed method. The first dataset was the U.S. trade import dataset that has been referenced in previous studies for trade product classification (Luppés, 2019; Ruder, 2020; Shubham *et al.*, 2023; Spichakova and Haav, 2020a), and the second dataset was the India import dataset, which has been studied in the work of Harsani (Harsani *et al.*, 2020). Unlike the study in Chapter 3, which utilised all HS codes available, this study specifically selected product categories that were considered challenging to classify by traders. The selected product categories include products in the categories of electrical machinery, televisions, personal computers, and mechanical appliances, corresponding to HS code chapters 84 and 85. HS Chapters 84 (Machinery and Mechanical Appliances) and 85 (Electrical Machinery and Equipment) are widely recognised as among the most challenging categories for product classification task (Lee *et al.*, 2023; Spichakova and Haav, 2020b). This complexity arises from their high semantic density and overlapping functional descriptions, as products are frequently characterised using highly technical and specialised terminology. Moreover, similar textual descriptions may correspond to different HS subheadings depending on functional attributes, level of integration, or intended use, thereby increasing intra-class similarity and inter-class ambiguity. These characteristics make Chapters 84 and 85 particularly appropriate for evaluating the robustness and discriminative capability of the proposed methods.

The distribution of trade product categories was imbalanced, with majority-to-minority class ratios of approximately 30:1. Text preprocessing was applied with lowercasing and punctuation removal to keep potentially valuable product information. Details of the datasets are provided in Table 4.1, where one dataset portion was allocated for training sentence embeddings, while the remaining data was used for training and validation of the classifiers through k-fold CV.

Datasets	Number of Classes	Number of Transactions	Sentence Embeddings	Classifiers
Datasets 1 (India dataset)	172	66,522	46,565	19,957
Datasets 2 (The U.S. dataset)	112	58,003	39,612	18,391

Table 4.1 The statistics of India and U.S. trade datasets available for the experiments, limited to HS Chapters 84 and 85.

4.3.2 Model Configuration

This section describes the configuration of components involved in the network architecture.

Backbone Encoders

Pretrained transformer-based models were sourced from the model hub² to serve as the foundation for the text encoder. We chose the DistilBERT and BERT language models, specifically the base-uncased variants, to ensure compatibility with the dataset's language, as these models have been pretrained on English corpora and have proven effective with trade data in previous experiments in Chapter 3.

Pooling Strategy

This thesis examined three pooling strategies for the network architecture for enhanced sentence embedding generation, including mean pooling, max pooling, and special token representation [CLS], to assess which strategy contributes to the enhancement of the embedding quality. The sentence embeddings dimensionality was set to 768, allowing for efficient application in evaluation and downstream tasks, including sentence embeddings quality measurement and text classification tasks.

Pairwise Samples

The training samples were arranged via a procedure to generate pairs of anchor positive samples for model training, with a positive sample being randomly selected without replacement to avoid duplication and to ensure optimal diversity. The training sample configuration employed one positive sample per anchor, but an extended

² <http://huggingface.co/>

examination was set to three positive samples to observe the effect on the generated embeddings evaluated on classification performance. Samples were organised into batches, with batch sizes of 8 and 32 observed to confirm the positive effect of large batch size in contrastive learning (Chen *et al.*, 2020).

Loss Functions Configuration

The loss function utilised three key parameter configurations involving scale value, similarity function types, and loss function variants. Due to limited systematic research on selecting optimal scaling factors, the baseline scale parameter was set to 20 (Henderson *et al.*, 2020; Reimers and Gurevych, 2019) and gradually increased in the experiment to assess its impact on model performance. Both cosine similarity and dot product were implemented to evaluate their relative performance on document length, especially the short length characterising trade data. In this case, cosine similarity provides a benefit by normalising vector magnitude and performs better with variable-length texts, while dot product preserves magnitude information (Thakur, Reimers, Rüklé, *et al.*, 2021). Additionally, an alternative loss function was employed, Symmetric Multiple Negative Ranking (SMNR), which advances standard MNR by computing bidirectional contrastive loss. Unlike asymmetric variants that consider anchor-to-positive relationships, SMNR loss averages the standard MNR computation over both forward pairings (anchor, positive) and backwards pairings (positive, anchor).

Hyperparameters Setting

A variety of hyperparameters were introduced during training, including batch size, learning rate, and epochs. These configuration settings were tested to assess their impact on model efficacy in generating sentence embeddings, a strategy commonly employed for optimisation purposes (Weerts *et al.*, 2020). To achieve an optimal sentence embedding used for the classification task, we systematically configure hyperparameters including batch sizes of 8 and 32, training epochs ranging from 10 to 30, and learning rates of 2×10^{-5} to 5×10^{-5} with the AdamW optimiser throughout all experiments.

Machine Learning Classifier

The quality of sentence embeddings was assessed by their implementation on a

classification task, where the enhanced sentence-level embeddings of product descriptions generated by the proposed models serve as input features for the classifiers. The underlying hypothesis states that enhanced-quality embeddings will simplify the classifier's learning task, resulting in improved classification accuracy. For this reason, two traditional machine learning classifiers were employed consisting of Support Vector Machine (SVM) and Random Forest (RF) (Breiman, 2001). These classifiers were selected because they are widely adopted for classification tasks (Fernández-Delgado *et al.*, 2014), and we expect that enhanced-quality embeddings will enable even traditional algorithms to achieve strong results. Implementation of SVM used scikit-learn with default parameters to handle nonlinear classification through kernel methods (Zareapoor *et al.*, 2018). RF was chosen as an ensemble approach for its effectiveness to handle high-dimensional and noisy text data (Islam *et al.*, 2019).

4.3.3 Model Baselines

For performance benchmarking, we compare our proposed method with the following language models:

- USE (Cer *et al.*, 2018) is a pretrained model designed to generate sentence-level embeddings, available in various architectures, including USE-transformer, which employs the Transformer architecture, with the use-cmlm-multilingual model used in this experiment.
- SBERT (Reimers and Gurevych, 2019) is a widely recognised sentence embedding framework that produces high-quality representations through bi-encoder architecture training with pretrained transformer models. This study utilises two SBERT variants consisting of nli-distilbert-base and nli-bert-base.
- BERT (Devlin *et al.*, 2019) and DistilBERT (Sanh *et al.*, 2020) were selected as strong baselines based on their superior performance over other transformer-based models on trade datasets in Chapter 3. For this purpose, both models were fine-tuned on domain-specific trade data with sentence representations extracted from the [CLS] token.

4.3.4 Experiments Evaluation

Two evaluation approaches were conducted to assess our proposed method. First,

the embeddings were evaluated based on the quality of improvements in structural representation and characteristics. Second, classification performance evaluated the model's performance when using enhanced sentence embeddings on classification tasks compared to the baseline models.

Embeddings Visualisation and Similarity

A straightforward approach to examine latent structural patterns and clustering behaviour involved visualisation via dimensionality reduction, which projects high-dimensional embeddings into a low-dimensional space. For this purpose, we employed Principal Component Analysis (PCA) (Jolliffe and Cadima, 2016) that preserves global data structure by maintaining distributional characteristics and t-Distributed Stochastic Neighbour Embedding (t-SNE) (Maaten and Hinton, 2008) that specialises in preserving local neighbourhood relationships during dimensionality reduction. Both methods transform high-dimensional embedding vectors into interpretable two-dimensional visualisations.

We also assessed the embedding quality by measuring the anisotropy score utilising similarity analysis. Embedding anisotropy was evaluated by computing the average cosine similarity between individual embeddings and the mean embedding vector for each model. Given a dataset $\mathcal{D}_{test} = \{(x_i, y_i)\}_{i=1}^n$ where x_i are embeddings generated by the model and y_i are class labels, we first computed the mean vector: $\mu = \frac{1}{n} \sum_{i=1}^n x_i$. The anisotropy score was then calculated as: $\text{score} = \frac{1}{n} \sum_{i=1}^n \text{sim}(\mu \cdot x_i)$ where $\text{sim}(x_a, x_b) = \frac{x_a \cdot x_b}{|x_a||x_b|}$. A higher score reflects stronger anisotropy, underlining embeddings cluster in a narrow direction instead of spreading evenly in the embeddings space.

The embedding quality was evaluated through intra-class and inter-class similarity analysis to assess clustering effectiveness and class separability. The intra-class similarity $\bar{S}_{\text{intra}}$ measures within-class cohesion: $\bar{S}_{\text{intra}} = \frac{1}{|P_{\text{intra}}|} \sum_{(i,j) \in P_{\text{intra}}} \text{sim}(x_i, x_j)$, where $P_{\text{intra}} = \{(x_i, x_j) : y_i = y_j\}$ denotes the set of intra-class sample pairs. On the other hand, the inter-class similarity, $\bar{S}_{\text{inter}}$, quantifies between-class separation: $\bar{S}_{\text{inter}} = \frac{1}{|P_{\text{inter}}|} \sum_{(i,k) \in P_{\text{inter}}} \text{sim}(x_i, x_k)$, where $P_{\text{inter}} = \{(x_i, x_k) : y_i \neq y_k\}$ denotes inter-

class sample pairs. Effective embeddings should exhibit high intra-class similarity and low inter-class similarity, indicating well-separated clusters in the embedding space.

Four embedding extraction methods were employed in this study, as illustrated in Figure 4.4:

- a. The first method used a transformer-based model (e.g., BERT and DistilBERT) without fine-tuning approach, where sentence-level representations were obtained from the [CLS] token in the final hidden layer.
- b. The second method utilised off-the-shelf sentence embedding models, including USE and SBERT, from the Sentence-Transformers library, hosted on the Hugging Face model hub³.
- c. The third method utilised a transformer-based model fine-tuned on the trade dataset.
- d. The proposed method which generating enhanced sentence embeddings.

³ <https://huggingface.co/>

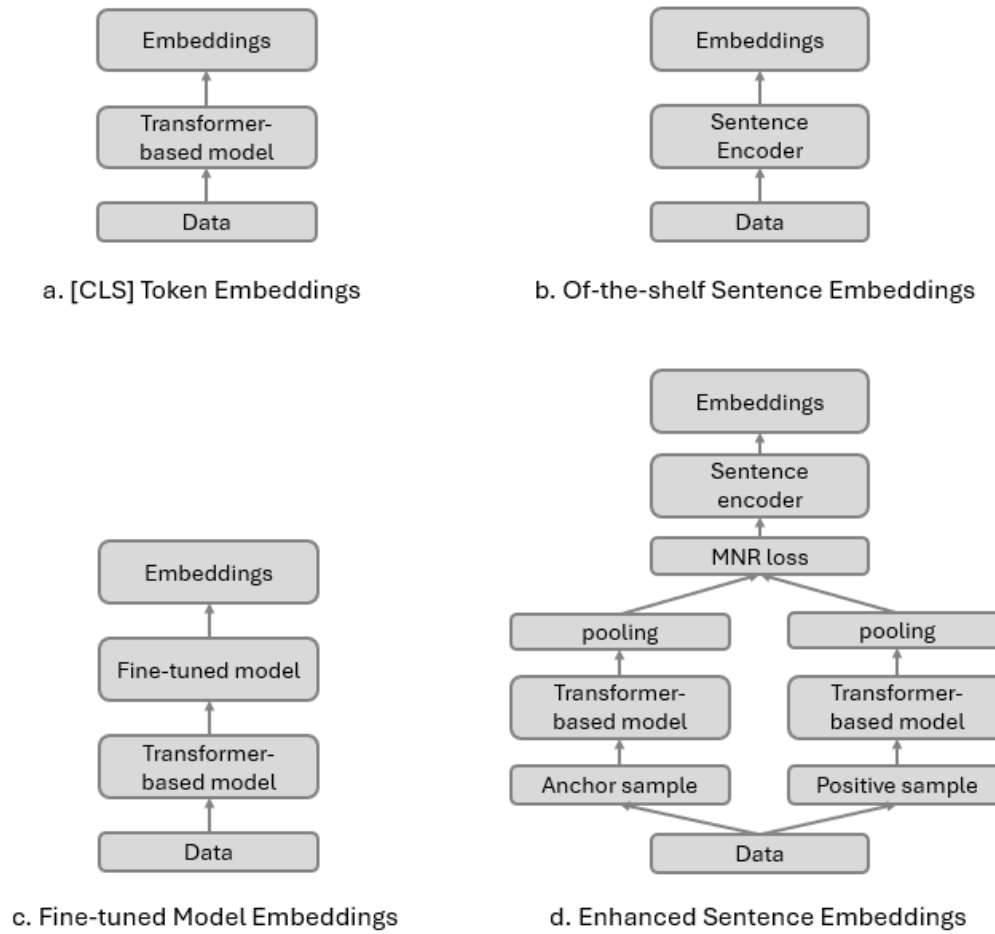


Figure 4.3 Several methods configured to generate sentence embedding method to represent sentence-level embeddings for product descriptions.

Classification Metrics

The models' performance was evaluated using precision, recall, and F1-score with weighted averaging to address class imbalance. Model generalization was assessed through 10-fold CV protocol (Kohavi, 1995), and Cohen's d was calculated to measure effect sizes when comparing performance between different models (Cohen, 1988).

4.4 Results and Discussion

The experimental results using the proposed method are arranged in two subsections. The first section presents the analysis of the intrinsic characteristics of the sentence embeddings, while the second presents the evaluation of their effectiveness in downstream classification tasks.

4.4.1 Embeddings Characteristics

A comparative analysis examined the geometric properties and representational quality of embeddings generated by each method to investigate the presence of anisotropy, with PCA used to visualise their distributions in embedding space. Figure 4 presents the experimental results for the India dataset only, as similar patterns were observed for the U.S. dataset. Utilising [CLS] token to represent sentence-level embeddings presented a tightly clustered circular distribution, with most data points concentrated in a confined area of the embedding space reflecting a high degree of anisotropy, as shown in Figure 4.4(a). Similarly, SBERT's sentence embeddings showed comparable clustering behaviour, with most vectors located in a limited region in the high-dimensional space, despite a slightly broader dispersion compared to the previous method, as seen in Figure 4.4 b. Fine-tuning approach partially addresses this issue, with generated embeddings presenting an improvement of spatial distribution, as shown in Figure 4.4 (c). Finally, using our proposed method, the sentence embeddings were more widely dispersed, with the distribution exhibiting clearer and more distinct clustering patterns, as presented in Figure 4.4 (d).

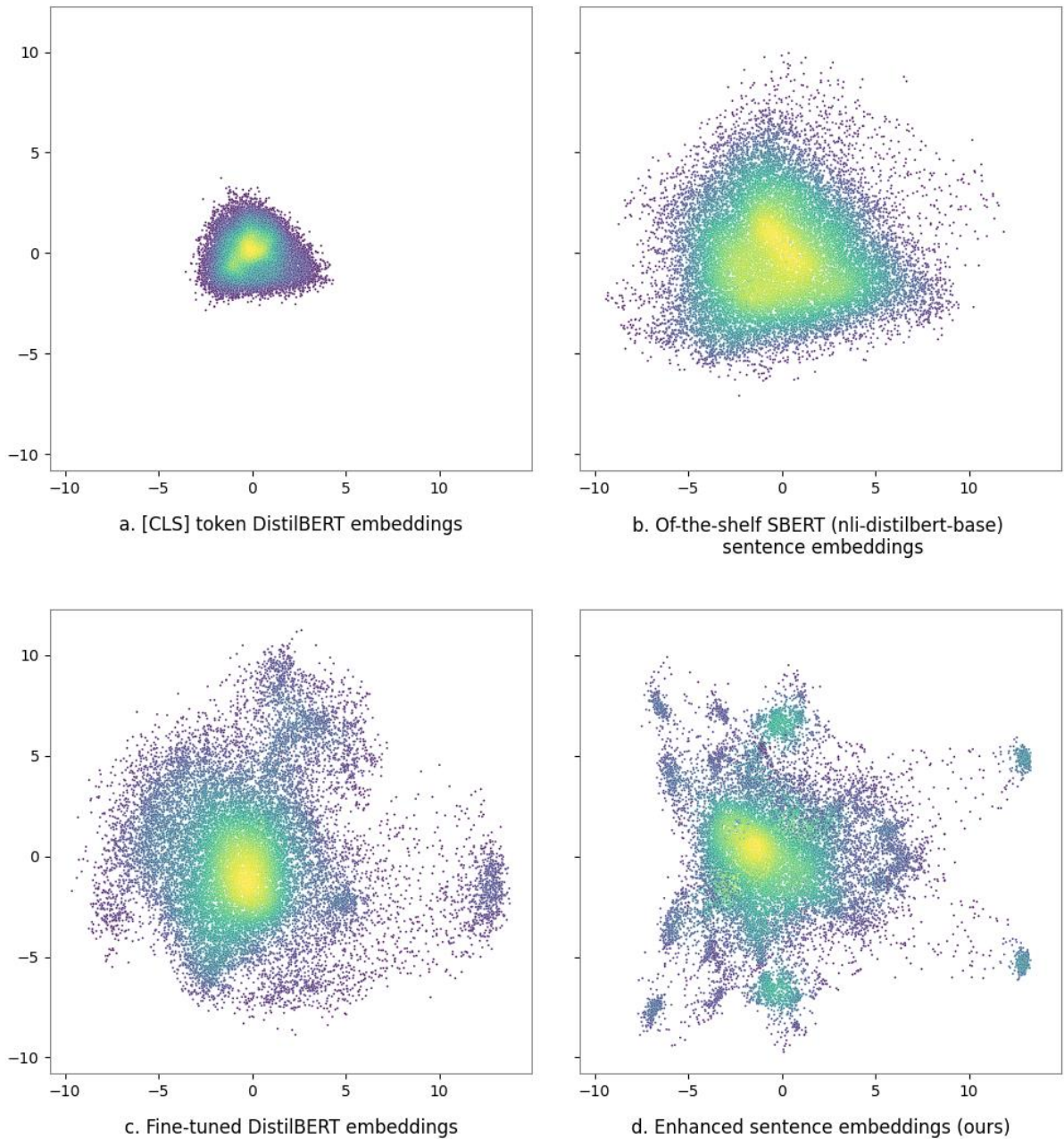


Figure 4.4 Visualisation of frequency-based distribution of generated sentence embeddings from each method. Lighter colours indicate regions of higher similarity frequency among embeddings.

Anisotropy was assessed through mean vector similarity score calculations, following the procedure in Section 4.3, applied to embeddings generated by each model. Without fine-tuning, the language model produced sentence embeddings with an anisotropy level of 0.97, while SBERT reduced this to 0.73, as shown in Figure 4.5 a and Figure 4.5 b. Both methods showed a concentration of data points around the mean, consistent with patterns previously reported in the literature (Ethayarajh, 2019;

Godey *et al.*, 2024). Trade data was used to fine-tune a transformer-based model that substantially mitigated the anisotropy issue, reducing the anisotropy score to 0.40, as seen in Figure 4.5 c. Our proposed model achieved the lowest score of 0.16, as shown in Figure 4.5 d. This lowest score corresponded to a more uniform distribution across embedding dimensions, underlining a considerable improvement in sentence representation to support discriminative power for the classification task.

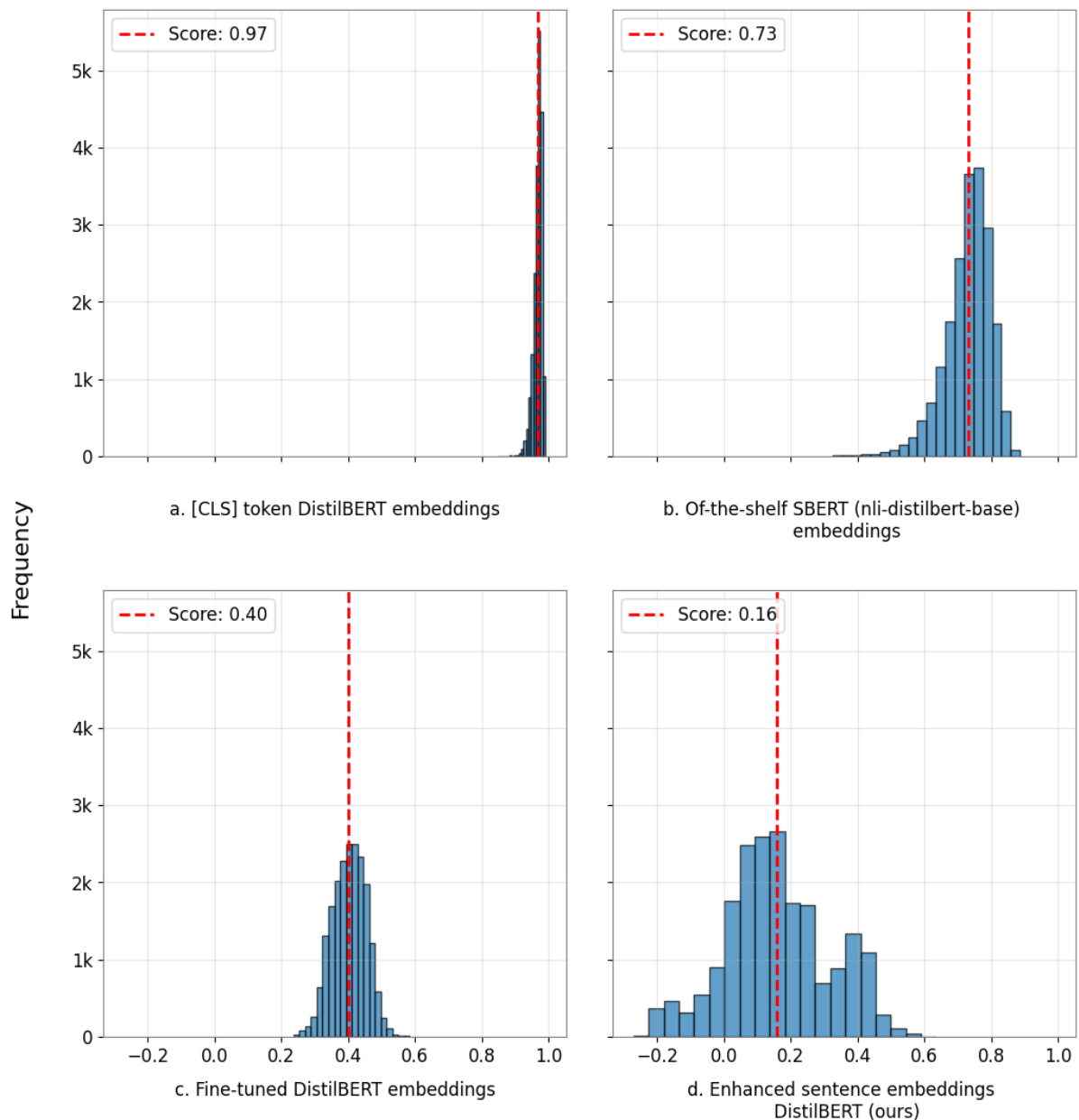


Figure 4.5 The average cosine similarity between the mean embedding vector and all individual embeddings in the dataset, accompanied by a visualisation of the similarity distribution. This analysis highlights the degree of anisotropy and embedding uniformity within the dataset.

T-SNE visualisation was employed to analyse class distribution and assess embedding discriminability across HS code categories, with each colour representing class labels based on HS code label. Models without a fine-tuning method generated scattered data points with substantial inter-class overlap, which limits their effectiveness in distinguishing between categories, as presented in Figure 4.6 (a) and (b). Additionally, data points in the same class category appeared dispersed rather than forming coherent clusters, which makes class boundaries difficult to distinguish visually. On the other hand, the fine-tuned model demonstrates improved class separation as illustrated in Figure 4.6 (c). This approach indicates enhanced discriminability of the sentence embeddings, which creates more apparent and compact clusters with transactions from similar categories to push more closely in the embedding space. Finally, our proposed method demonstrates significantly improved discriminability by generating more cohesive class clusters and greater inter-class separation in the embedding space, as illustrated in Figure 4.6 (d). This structure aligns with this finding (Khosla *et al.*, 2021), who demonstrated that supervised contrastive learning pulls samples from the same class closer together while pushing apart those from different classes, thereby producing more discriminative features and facilitating simpler decision boundaries. The presented visualisation implies that our proposed method effectively aligns related instances while maximising separation between classes, improving the overall embedding structure.

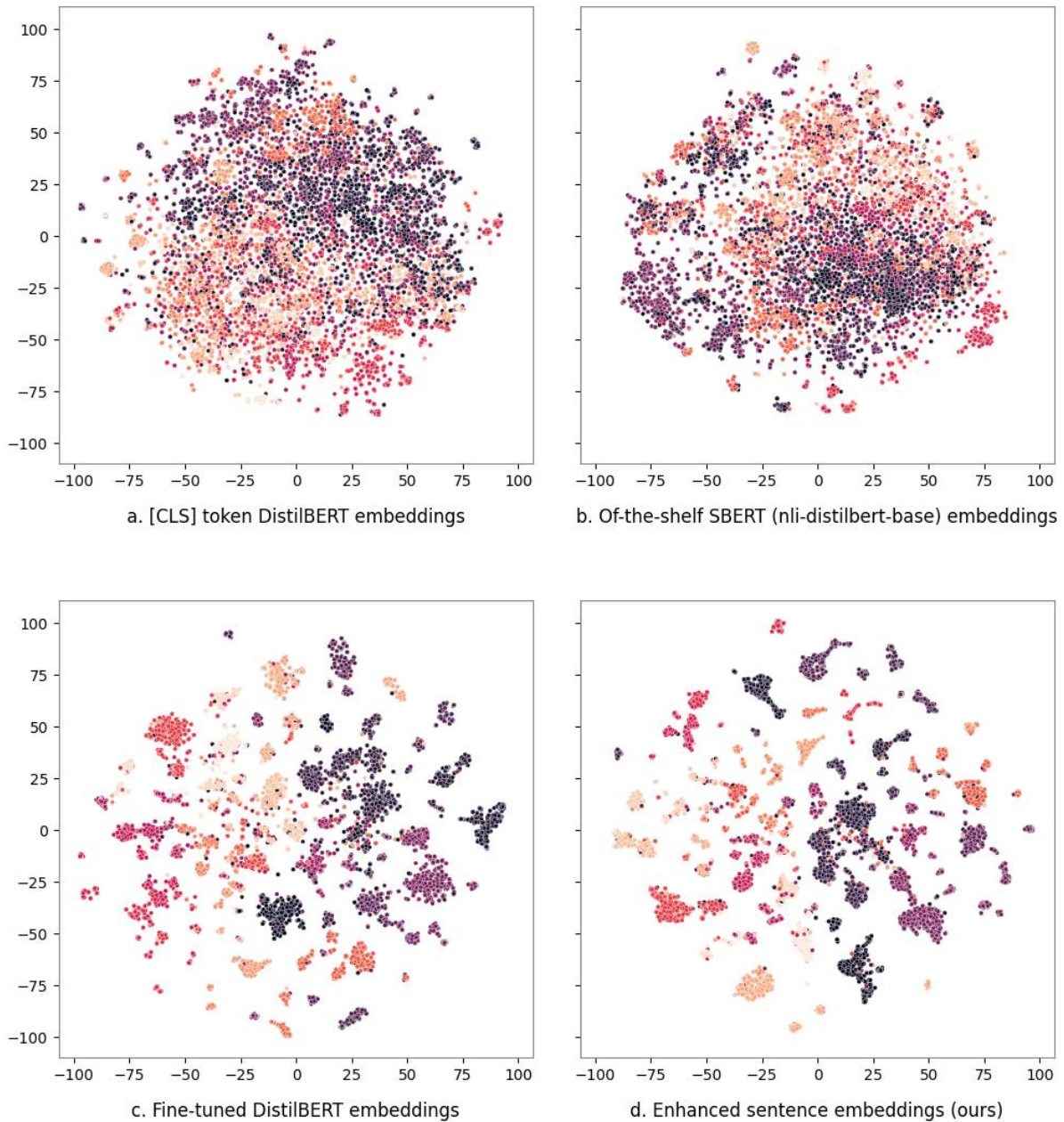


Figure 4.6 The figure illustrates the distribution and clustering behaviour of embeddings across methods, highlighting the representational structure and separability of trade data within similar product categories in embedding space.

Visual analysis found that our proposed method generated enhanced geometric transformations of sentence embeddings in the embedding space. Table 4.2 presents the improvement of similarity scores comprising intra-class similarity (products within the same category) and inter-class similarity (products from different categories). Our proposed method achieved two key improvements over the fine-tuned baseline by increasing intra-class similarity, which pulls related products closer together, and

decreasing inter-class similarity, improving separation between distinct product categories, as shown in Table 4.2. These results indicate that the improved clustering structure, reflected in both intra- and inter-class similarity, supports our hypothesis that contrastive learning optimises embedding geometry, yields stronger decision boundaries, and is expected to improve downstream classification performance.

Extraction Method	Similarity Score	
	Intra-class	Inter-class
Baseline method:		
a. [CLS] token DistilBERT embeddings	0.95	0.94
b. Of-the-shelf SBERT (nli-distilbert-base)	0.61	0.53
c. Fine-tuned DistilBERT embeddings	0.70	0.16
Proposed method:		
d. Enhanced sentence embeddings (DistilBERT)	0.81	0.02

Table 4.2 Comparison of similarity scores between intra-class and inter-class generated from each method.

4.4.2 Classification Performance

This section shows the experimental results of trade product classification between the proposed method and several baseline models.

Our proposed method outperformed all baseline models across both datasets. It achieved the highest F1-score of 86.5% for Dataset 1, as reported in Table 4.3, and 80.2% for Dataset 2, as shown in Table 4.5. On average, these findings represent an improvement of more than 20% over the baseline model and approximately 3% over the strongest fine-tuned model. Among the baseline models, the USE demonstrated superior performance compared to the SBERT variants. Meanwhile, fine-tuned transformer-based models yield improved classification results, in line with evidence reported in Chapter 3. However, despite these enhancements, their performance still falls short of that achieved by the proposed method. The statistical analysis for Dataset 1 is summarised in Table 4.6, and for Dataset 2 is presented in Table 4.6, confirming that our proposed methods are statistically significant with a large effect size.

Another finding in our strategy of using different underlying base encoders (e.g., BERT vs. DistilBERT) for generating sentence embeddings shows no statistically significant

performance differences within the proposed model architecture. This finding indicates that the observed improvements are attributable to the contrastive learning methodology rather than the specific choice of transformer-based encoder. Additionally, the results underlined a notable finding that our proposed method achieved strong classification performance when paired with traditional machine learning algorithms, such as SVM and RF. This approach demonstrates the flexibility and generalisability of the generated enhanced sentence embeddings, which can be applied across different classifiers and datasets, surpassing the performance of fine-tuned transformer models with a neural network head classifier.

Dataset	Method	Precision	Recall	F1-score
1	<i>Baseline models:</i>			
	USE _{SVM}	0.740±0.008	0.708±0.009	0.697±0.009
	USE _{RF}	0.690±0.010	0.626±0.009	0.609±0.010
	SBERT _{nli-distilbert-base + SVM}	0.625±0.013	0.592±0.009	0.571±0.011
	SBERT _{nli-distilbert-base + RF}	0.635±0.008	0.564±0.008	0.549±0.008
	SBERT _{nli-bert-base + SVM}	0.640±0.010	0.601±0.005	0.584±0.004
	SBERT _{nli-bert-base + RF}	0.643±0.014	0.583±0.010	0.568±0.010
	Fine-tuning _{DistilBERT}	0.845±0.010	0.840±0.009	0.831±0.009
	Fine-tuning _{BERT}	0.843±0.013	0.844±0.010	0.832±0.011
	<i>Proposed methods:</i>			
	Sentences Emb. _{DistilBERT + SVM}	0.870±0.004	0.862±0.006	0.860±0.006
	Sentences Emb. _{DistilBERT + RF}	0.872±0.005	0.864±0.005	0.862±0.006
	Sentences Emb. _{BERT + SVM}	0.872±0.005	0.863±0.005	0.862±0.005
	Sentences Emb. _{BERT + RF}	0.875±0.006	0.867±0.006	<u>0.865±0.006</u>

Table 4.3 Comparison of the model classification performance on Dataset 1, demonstrating enhanced sentence embeddings combined with SVM and RF classifiers outperforming baseline models across key evaluation metrics.

Dataset	Models	p-value	Wilcoxon	Mean Difference	Effect Size (Cohen's <i>d</i>)
1	USE _{SVM}	<0.01	<0.01	0.17	22.19
	USE _{RF}	<0.01	<0.01	0.26	32.42
	SBERT _{nli-distilbert-base + SVM}	<0.01	<0.01	0.29	34.44
	SBERT _{nli-distilbert-base + RF}	<0.01	<0.01	0.32	44.58
	SBERT _{nli-bert-base + SVM}	<0.01	<0.01	0.28	57.28
	SBERT _{nli-bert-base + RF}	<0.01	<0.01	0.30	36.36
	Fine-tuning _{DistilBERT}	<0.01	<0.01	0.08	14.96
	Fine-tuning _{BERT}	<0.01	<0.01	0.08	15.98

Table 4.4 Statistical comparison between the best-performing configuration of the proposed method and baseline models on Dataset 1, presenting that the proposed method significantly outperformed the baseline methods, with statistically significant differences observed.

Dataset	Method	Precision	Recall	F1-score
2	<i>Baseline models:</i>			
	USE _{SVM}	0.675±0.016	0.630±0.015	0.625±0.015
	USE _{RF}	0.594±0.015	0.553±0.015	0.538±0.014
	SBERT _{nli-distilbert-base + SVM}	0.591±0.016	0.546±0.014	0.535±0.014
	SBERT _{nli-distilbert-base + RF}	0.571±0.010	0.531±0.011	0.517±0.011
	SBERT _{nli-bert-base + SVM}	0.593±0.016	0.546±0.013	0.536±0.012
	SBERT _{nli-bert-base + RF}	0.563±0.012	0.531±0.007	0.516±0.007
	Fine-tuning _{DistilBERT}	0.788±0.011	0.783±0.010	0.774±0.011
	Fine-tuning _{BERT}	0.801±0.012	0.798±0.013	0.788±0.012
	<i>Proposed methods:</i>			
	Sentences Emb. DistilBERT + SVM	0.808±0.011	0.795±0.012	0.794±0.011
	Sentences Emb. DistilBERT + RF	0.810±0.011	0.797±0.012	0.797±0.012
	Sentences Emb. BERT + SVM	0.813±0.011	0.802±0.010	0.802±0.010
	Sentences Emb. BERT + RF	0.813±0.014	0.802±0.014	<u>0.802±0.013</u>

Table 4.5 Comparison of the model classification performance on Dataset 2, demonstrating enhanced sentence embeddings combined with SVM and RF classifiers outperforming baseline models across key evaluation metrics.

Dataset	Models	p-value	Wilcoxon	Mean Difference	Effect Size (Cohen's <i>d</i>)
2	USE _{SVM}	<0.01	<0.01	0.18	12.53
	USE _{RF}	<0.01	<0.01	0.26	19.24
	SBERT _{nli-distilbert-base + SVM}	<0.01	<0.01	0.27	19.72
	SBERT _{nli-distilbert-base + RF}	<0.01	<0.01	0.29	23.28
	SBERT _{nli-bert-base + SVM}	<0.01	<0.01	0.27	20.63
	SBERT _{nli-bert-base + RF}	<0.01	<0.01	0.29	26.95
	Fine-tuning _{DistilBERT}	<0.01	<0.01	0.03	2.37
	Fine-tuning _{BERT}	<0.05	<0.05	0.01	1.10

Table 4.6 Statistical comparison between the best-performing configuration of the proposed method and baseline models on Dataset 2, presenting that the proposed method significantly outperformed the baseline methods, with statistically significant differences observed.

Next, wall-clock training duration was measured to evaluate the relationship between performance improvements and computational cost to indicate practicality and scalability for rapid deployment scenarios (Raihan *et al.*, 2019).

Runtime analysis was conducted for SVM and RF classifiers across models. It was evident that RF required longer training times than SVM when using sentence embeddings generated from off-the-shelf language models such as USE and SBERT variants. RF required approximately twice the training time of SVM for baseline models, as shown in Figure 4.7. When applying our proposed method, RF maintained longer training times than SVM, with this computational overhead being more pronounced than with baseline embeddings. Conversely, SVM achieved improved efficiency, where the training time is approximately one-third faster than with pretrained sentence embeddings. Contrastive learning produced sentence embeddings that present a more linearly separable representation in the feature space, allowing SVM to perform more effectively. This approach different with RF, which functions on multiple decision trees to handle nonlinear separability underlying representation learning, shifting computational trade-offs between algorithms.

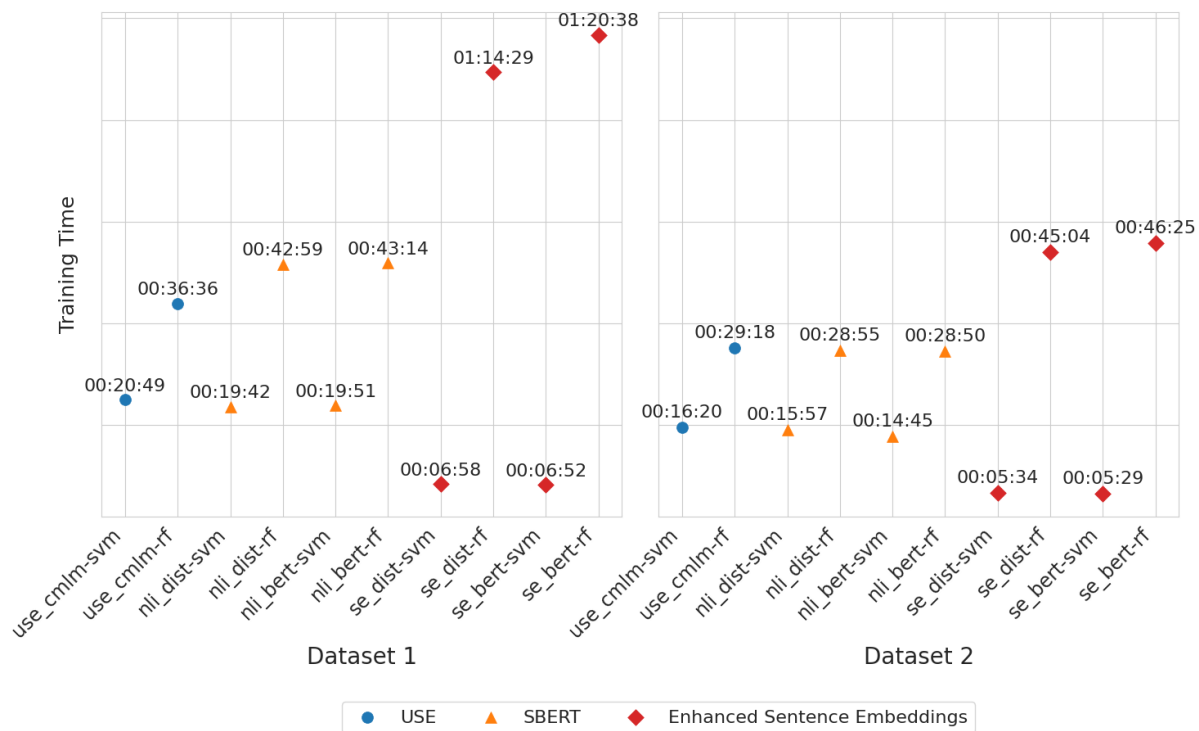


Figure 4.7 Duration of training time (hours: minutes: seconds) across two datasets using USE, SBERT, and the proposed method using SVM and RF classifiers (trained on an NVIDIA Tesla P100 GPU).

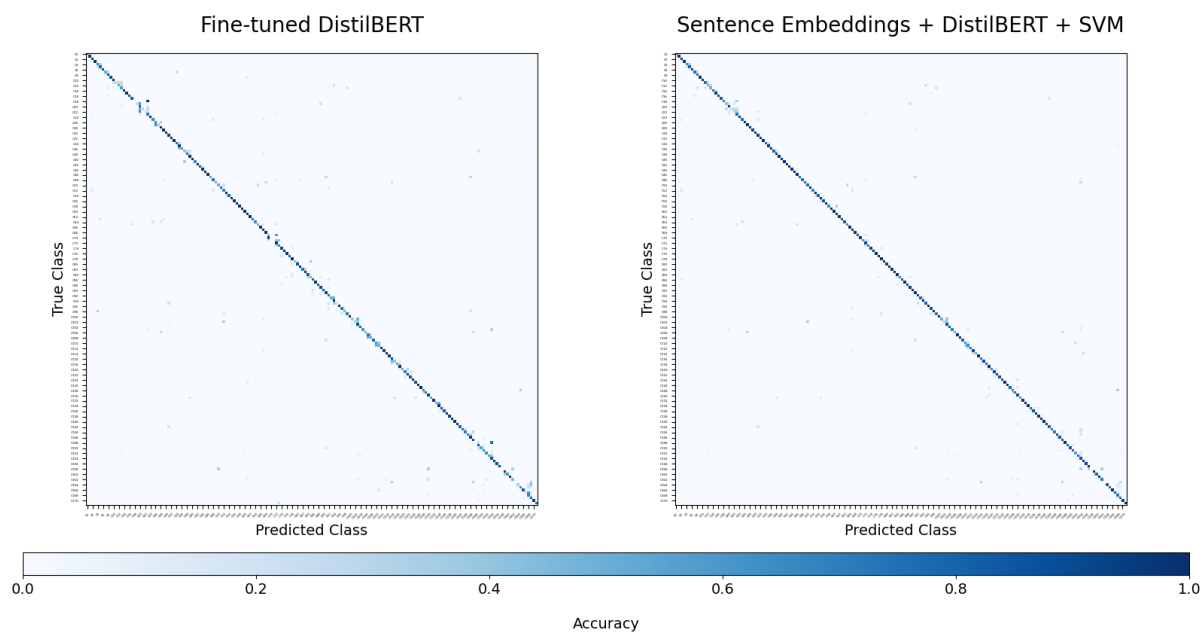


Figure 4.8 Confusion matrix comparison between the fine-tuned DistilBERT model and the proposed method.

The evaluation result of classification models performance for both the fine-tuned

model and our proposed method was presented through confusion matrix analysis, which provides a summary of correctly classified and misclassified instances across 172 classes in Dataset 1, as illustrated in Figure 4.8. Prediction accuracy is visualised using colour intensity, with values ranging from 0.0 to 1.0 and darker blue indicates higher F1-scores, while lighter blue indicates lower F1-scores and the gap of selected categories between the two approaches is detailed in Table 4.7. Markedly, certain product categories, particularly class 71 (HS code 84818030), exhibited performance enhancement under our proposed method, achieving a F1-score of 75.00%, but 0% F1-score with the fine-tuning approach.

Class	F1-score	
	Fine-tuned DistilBERT Model	Sentences Emb. + DistilBERT + SVM
Class-71	0.000	0.750
Class-167	0.333	0.889
Class-69	0.429	0.923
Class-95	0.333	0.800
Class-108	0.571	1.000

Table 4.7 Classes with the substantial F1-score differences between the fine-tuned DistilBERT model and the proposed method.

For example, the product "*kitz industrial valves- brass ball valve (phtr)sza3/8 (1200 pcs @ usd 1.75 per pc)*" was correctly classified by the proposed method as HS code 84818030. According to HS code guidance (World Customs Organization, 2018b), this category belongs to valves intended for general fluid flow control in pipes or tanks and thus is suitable for general-purpose applications. However, the product category predicted utilising the fine-tuned model was incorrect, as it was assigned to HS code 84812000, a category designated for valves used in oleohydraulic or pneumatic transmission, which does not accurately reflect the given product description.

4.4.3 Components Evaluation

This subsection provides an evaluation of the model components based on the generation of the sentence embedding, consisting of three main aspects: the number of paired training samples, the choice of pooling strategies, and the configuration of loss functions. All the components to generate sentence embeddings were evaluated

based on the model's performance in classifying the product category.

Samples configuration

Changing the number of positive samples paired with an anchor has a significant impact on model performance. In this experiment, the number of positive samples was set to three, with results that are consistently observed across both datasets, as shown in Table 4.8.

Dataset	Samples	Precision	Recall	F1-score
1	Sentences Emb. DistilBERT + SVM			
	1 anchor, 1 positive	0.848±0.006	0.840±0.004	0.836±0.005
	1 anchor, 3 positive	0.870±0.004	0.862±0.006	<u>0.860±0.006*</u>
	Sentences Emb. DistilBERT + RF			
	1 anchor, 1 positive	0.853±0.005	0.845±0.005	0.841±0.005
	1 anchor, 3 positive	0.872±0.005	0.864±0.005	<u>0.862±0.006*</u>
2	Sentences Emb. DistilBERT + SVM			
	1 anchor, 1 positive	0.788±0.011	0.775±0.013	0.773±0.013
	1 anchor, 3 positive	0.808±0.011	0.795±0.012	<u>0.794±0.011*</u>
	Sentences Emb. DistilBERT + RF			
	1 anchor, 1 positive	0.785±0.011	0.773±0.012	0.772±0.012
	1 anchor, 3 positive	0.810±0.011	0.797±0.012	<u>0.797±0.012*</u>

Table 4.8 Comparison of the model classification performance under different numbers of positive samples per anchor.

Pooling

In the experiments, we compared different pooling strategies: mean, max, and the [CLS] token. As summarised in Table 4.9, the results indicated that no single strategy consistently yields superior outcomes across all experiments. This lack of a dominant pooling approach underlines the unavailability of a clear distinction among the pooling strategies concerning their impact on model performance.

Dataset	Pooling Strategy	Precision	Recall	F1-score
1	Sentences Emb. DistilBERT + SVM			
	CLS	0.871±0.006	0.862±0.006	0.860±0.006
	Mean	0.870±0.004	0.862±0.006	0.860±0.006
	Max	0.873±0.006	0.864±0.007	0.863±0.007
	SE (DistilBERT + RF)			
	CLS	0.868±0.004	0.858±0.004	0.857±0.004
	Mean	0.872±0.005	0.864±0.005	0.862±0.006
	Max	0.871±0.004	0.862±0.004	0.861±0.004
2	Sentences Emb. DistilBERT + SVM			
	CLS	0.813±0.015	0.799±0.015	0.800±0.015
	Mean	0.808±0.011	0.795±0.012	0.794±0.011
	Max	0.809±0.009	0.794±0.012	0.795±0.011
	SE (DistilBERT + RF)			
	CLS	0.811±0.013	0.800±0.013	0.800±0.013
	Mean	0.810±0.011	0.797±0.012	0.797±0.012
	Max	0.809±0.011	0.796±0.013	0.796±0.012

Table 4.9 Comparison of the model classification performance when using three pooling strategies, including CLS token, mean pooling, and max pooling.

Loss function

Table 4.10 summarises the results of different loss function configurations when developing sentence embedding models on the downstream classification task performance. Adjusting the scale parameter from 20 to 50 did not improve the model performance, underlining that the model shows a low sensitivity to this parameter.

Replacing cosine similarity with the dot product led to a statistically significant reduction in F1-score when SVM was used as the classifier. The observed decline in performance appears to be related to the use of dot-product loss in sentence embeddings, which can fluctuate in magnitude. Since SVM focuses on optimising distances by maximising the margin (Cortes and Vapnik, 1995), it can be particularly affected by variations in embedding scales. It tends to perform inadequately with unnormalised dot product embeddings but shows improvement with cosine similarity because of its inherent normalisation. On the other hand, RF utilised scale-invariant,

tree-based decision splits, indicated robustness to such variations and maintains more stable classification performance.

SMNR loss achieved F1-scores similar to MNR loss, showing that both are equally suitable for training.

Dataset	Loss Function Parameters	Precision	Recall	F1-score
1	Sentences Emb. DistilBERT + SVM			
	MNR Loss:			
	Scale – 20 (CosSim)	0.870±0.004	0.862±0.006	0.860±0.006
	Scale – 50 (CosSim)	0.869±0.004	0.860±0.004	0.858±0.004
	Dot Product	<u>0.521±0.007</u>	<u>0.642±0.004</u>	<u>0.555±0.004</u>
	Symmetric MNR Loss	0.872±0.005	0.863±0.005	0.861±0.006
	Sentences Emb. DistilBERT + RF			
	MNR Loss:			
	Scale – 20 (CosSim)	0.872±0.005	0.864±0.005	0.862±0.006
	Scale – 50 (CosSim)	0.872±0.003	0.864±0.003	0.863±0.003
	Dot Product	0.868±0.007	0.860±0.007	0.858±0.007
	Symmetric MNR Loss	0.874±0.006	0.865±0.005	0.864±0.006
2	Sentences Emb. DistilBERT + SVM			
	MNR Loss:			
	Scale – 20 (CosSim)	0.808±0.011	0.795±0.012	0.794±0.011
	Scale – 50 (CosSim)	0.808±0.011	0.794±0.013	0.794±0.012
	Dot Product	<u>0.565±0.010</u>	<u>0.642±0.009</u>	<u>0.580±0.007</u>
	Symmetric MNR Loss	0.811±0.011	0.800±0.012	0.799±0.012
	Sentences Emb. DistilBERT + RF			
	MNR Loss:			
	Scale – 20 (CosSim)	0.810±0.011	0.797±0.012	0.797±0.012
	Scale – 50 (CosSim)	0.809±0.010	0.795±0.012	0.795±0.012
	Dot Product	0.808±0.013	0.797±0.013	0.796±0.013
	Symmetric MNR Loss	0.809±0.010	0.799±0.011	0.799±0.011

Table 4.10 Comparison of the model classification performance across varied loss function configurations.

Hyperparameter Optimisation

To enhance the quality of the sentence embeddings, we performed hyperparameter

tuning by adjusting the batch size, learning rate, and number of training epochs. The refined model through this process was subsequently used to generate sentence embeddings, which were evaluated using machine learning classifiers to determine their effect on overall model performance. As depicted in Table 4.11, the hyperparameter selection utilised random search and tested various configurations, including changes in batch size, learning rate, and number of epochs.

The experimental results showed that increasing the batch size to 32 produced a notable improvement in model performance. However, statistical significance was observed only in the case of using SVM as the classifier for Dataset 1. As such, while the use of a larger batch size appears to enhance the quality of the sentence embeddings and improve downstream classification performance, the generalisability of this effect across different classifiers and datasets remains limited. Training with a larger number of epochs consistently improved performance utilising both datasets and all classifiers, underlining that extended training produces statistically significant classification results. Learning rate optimisation yielded an improvement in classification performance, showing that the model with a lower learning rate outperformed a higher learning rate. However, the model showed significant gains only on Dataset 1, underlining its limited generalisability.

Dataset	Loss Function Parameters	Precision	Recall	F1-score
1	Sentences Emb. DistilBERT + SVM			
	Batch 8	0.870±0.004	0.862±0.006	0.860±0.006
	Batch 32	0.876±0.006	0.867±0.005	<u>0.865±0.006*</u>
	Epoch 10	0.851±0.004	0.843±0.004	0.839±0.004
	Epoch 30	0.870±0.004	0.862±0.006	<u>0.860±0.006*</u>
	LR 2x10 ⁻⁵	0.870±0.004	0.862±0.006	<u>0.860±0.006*</u>
	LR 5x10 ⁻⁵	0.859±0.006	0.851±0.007	0.848±0.006
	Sentences Emb. DistilBERT + RF			
	Batch 8	0.872±0.005	0.864±0.005	0.862±0.006
	Batch 32	0.877±0.005	0.869±0.005	0.867±0.005
	Epoch 10	0.858±0.006	0.850±0.005	0.847±0.006
	Epoch 30	0.872±0.005	0.864±0.005	<u>0.862±0.006*</u>
	LR 2x10 ⁻⁵	0.872±0.005	0.864±0.005	<u>0.862±0.006*</u>
	LR 5x10 ⁻⁵	0.858±0.005	0.850±0.005	0.848±0.005
2	Sentences Emb. DistilBERT + SVM			
	Batch 8	0.808±0.011	0.795±0.012	0.794±0.011
	Batch 32	0.811±0.006	0.796±0.006	0.797±0.005
	Epoch 10	0.794±0.010	0.781±0.012	0.779±0.012
	Epoch 30	0.808±0.011	0.795±0.012	<u>0.794±0.011*</u>
	LR 2x10 ⁻⁵	0.808±0.011	0.795±0.012	0.794±0.011
	LR 5x10 ⁻⁵	0.804±0.011	0.791±0.012	0.791±0.011
	Sentences Emb. DistilBERT + RF			
	Batch 8	0.810±0.011	0.797±0.012	0.797±0.012
	Batch 32	0.811±0.008	0.798±0.009	0.798±0.008
	Epoch 10	0.797±0.012	0.786±0.011	0.785±0.012
	Epoch 30	0.810±0.011	0.797±0.012	<u>0.797±0.012*</u>
	LR 2x10 ⁻⁵	0.810±0.011	0.797±0.012	0.797±0.012
	LR 5x10 ⁻⁵	0.800±0.010	0.789±0.010	0.789±0.009

Table 4.11 Comparison of the model classification performance under different hyperparameter configurations reporting evaluation metrics across varying learning rates, batch sizes, and epochs.

4.5 Summary

This study addresses the second research objective, focused on improving the quality of sentence embeddings as a representation of product descriptions. The models were trained following a supervised contrastive learning approach by integrating multiple key components designed to enhance sentence embeddings to represent trade transactions with identical product categories to close together within a shared embedding space. The proposed approach utilised the generated sentence embeddings as input features for traditional machine learning classifiers, achieving higher performance than well-established fine-tuned transformer-based models, showing empirical evidence of the effectiveness of the proposed method. The proposed approach offers flexibility by allowing the high-quality sentence embeddings to be used with a variety of classifiers while reducing the training time required for developing classification models and improving the accuracy of product category predictions to provide a potential practical contribution.

The subsequent chapter will address product misclassifications in international trade transactions, with a particular focus on methodologies for identifying text anomaly detection.

Chapter 5

Text Anomaly Detection for Trade Product Misclassification via Autoencoders

This chapter addresses the third research objective to detect product misclassification in the trade domain by developing text anomaly detection. Product misclassification occurs when a product is assigned to an incorrect product code, which can be motivated by unlawful intentions. Fraudulent traders exploit the complexity of assigning product categories and submit ambiguous product descriptions, which results in misinterpretation when verifying the validity of the product category. In this chapter, anomaly detection was developed by focusing on detecting anomalies using product descriptions or text data, an element available in trade declarations. Given the complexity of product descriptions, text anomaly detection was developed employing autoencoders combined with modern language models.

5.1 Introduction

According to the WCO (World Customs Organization, 2018c), product misclassification is a component of tax and customs evasion in international trade, where traders intentionally provide inaccurate information in trade declarations to reduce their tax and duty obligations (Working Group on Commercial Fraud, 2010). Product misclassification also potentially causes significant threats to the integrity and functionality of tax and customs administration (Lushi and Gashi, 2016), leading to the under-collection of revenues from the trade domain (Anne *et al.*, 2023; Ding *et al.*, 2015; Global Financial Integrity, 2021; Judge, 2020; Office of The Auditor General of Canada, 2017). Apart from fiscal implications, undetected product misclassifications can impair trade data quality, diminish the reliability of analytical processes and hinder

evidence-based policy formulation (Grainger, 2024). Some factors motivate product misclassification, including selecting product categories that have lower tax or duty rates, or selecting products that have more benefits regarding quotas or trade agreements (Betz, 2019; Che *et al.*, 2025; Farhad *et al.*, 2024; Heijde, 2019; Kappler, 2011; Lombardi *et al.*, 2013). Multiple pieces of evidence of misclassification, which are linked to economic losses, are provided by some jurisdictions (Anne *et al.*, 2023; Ding *et al.*, 2015; European Court of Auditors, 2017; Global Financial Integrity, 2021; Judge, 2020; Office of The Auditor General of Canada, 2017).

Research studies in the trade domain have addressed the detection of tax and customs evasion activities through various approaches, which are covered in Chapter 2, discussing several methods for detecting fraudulent transactions. These methods range from rule-based systems (A. Kumar Singh *et al.*, 2003; Roman *et al.*, 2006), traditional machine learning models (Kim *et al.*, 2020; Vanhoeyveld *et al.*, 2020), and neural networks (Kunickaite *et al.*, 2021) that utilise diverse data types. Besides numerous previous research studies performed, they have not fully leveraged textual trade data, especially with recent advancements in NLP; text data has the potential to capture contextual nuances that reveal discrepancies between product descriptions and their assigned categories. A study addressing product misclassification using text data encoded by Doc2vec is considered insufficient to capture the nuanced semantic relationships present in textual descriptions (Spichakova and Haav, 2020b). Another study that utilised text elements from a trade dataset applied contextualised embeddings to represent product descriptions and employed an unsupervised learning strategy (Chen, Ben van Rijnsoever, *et al.*, 2021). However, the use of centroid vectors, obtained by averaging embeddings from historical records, was limited in its capacity to capture the subtle and complex characteristics of valid trade transactions.

Anomaly detection in text data is difficult due to linguistic complexity and the subtle nature of inconsistencies (Novoa-Paradela *et al.*, 2024). Product misclassification can be categorised as contextual anomalies, because the identification of anomalies depends on the surrounding text in product descriptions and the classification context in the product category. This characteristic makes them more challenging to identify than point anomalies (Chandola *et al.*, 2009), where individual data points clearly

deviate from the rest of the dataset. Chandola (Chandola *et al.*, 2009) defines contextual anomalies, also termed conditional anomalies, as characterised by two attribute sets: contextual attributes that provide a reference frame, and behavioural attributes that are evaluated against this context. Consider an electronic trade product described as "*27-inch LED monitor for a desktop computer*". The description refers to a monitor, but it may be perceived as a desktop computer. In this case, product description as a behavioural attribute is inconsistent with the assigned product category as a contextual attribute, which constitutes contextual anomalies.

The study of misclassification using text data within trade data remains underexplored. This task is situated within the domain of text anomaly detection, a relatively underexplored area in the field of NLP and compared to other domains of anomaly detection (Ait-Saada and Nadif, 2023; Novoa-Paradela *et al.*, 2024). Multiple factors contribute to the difficulty of detecting anomalies in text data. The primary consideration is the inherent linguistic complexity of text, posing modelling challenges such as the high dimensionality of text data (Barrett *et al.*, 2019; Li, Pei, *et al.*, 2023; Xu, Gábor, *et al.*, 2023). Another important factor for text anomaly detection is highly context-dependent, since anomalous text often varies across datasets and depends on the context. The lack of evaluation benchmarks also caused these limitations, which have recently been addressed in this literature (Li *et al.*, 2024). Chapter 2 of this thesis exposes that limited research studies have applied autoencoder-based methods for text anomaly detection. Despite of these challenges, there is opportunity to detect anomalies using text data with autoencoders, which have proven effective for anomaly detection (Chalapathy and Chawla, 2019) and still underutilised in text-based applications (Pang *et al.*, 2022).

Autoencoders are a class of deep neural networks that use reconstruction-based mechanisms. It consists of two main components, the encoder and the decoder. The process initiates with encoding the input data into a compressed low-dimensional representation known as a latent representation, performed by the encoder. The decoder then decodes the latent representation to reconstruct output that closely approximates the original input (Bank *et al.*, 2021; Hinton and Zemel, 1993). With this ability, autoencoders were originally developed for dimensionality reduction tasks by considering the latent representation as the reduced form of the original input data

(Fournier and Aloise, 2019; Hinton and Salakhutdinov, 2006). Nevertheless, the mechanisms inherent in these networks enable them to detect anomalies effectively. The learning process of autoencoders and their ability to create latent representations have proven effective for anomaly detection (Berahmand *et al.*, 2024; Hawkins *et al.*, 2002; Japkowicz *et al.*, 1995). The principle behind using autoencoders for anomaly detection operates as follows: when input data represents normal patterns, the autoencoder can reconstruct the output similarly to the original input, resulting in low reconstruction error. In contrast, anomalous data produce high reconstruction errors because the model has not learned these anomalous patterns during training, leading to the model's failure to reconstruct them accurately. The reconstruction errors can be used to set a threshold to identify whether new data represents normal patterns or anomalies. Research study in the field of anomaly detection using autoencoders has been applied in several studies employing various data types in different domains. However, the application of autoencoders for text anomaly detection is still in the early stages and has not been fully explored (Li, Pei, *et al.*, 2023; Novoa-Paradela *et al.*, 2024). Research in this field is lacking for the trade domain and limited in other domains, including anomaly detection in accident report documents (Rybak and Hassall, 2021) and social media content analysis on Twitter (Li *et al.*, 2021).

This thesis proposes text anomaly detection to manage the issue of product misclassification in international trade. In the context of this thesis, an anomaly refers to an observation that deviates from the patterns learned by the classification model during training. However, anomalies identified by the model do not necessarily represent errors, as their interpretations may vary. First, product misclassification refers to instances that deviate from the learned representation of correctly classified products and are flagged through high anomaly scores. Such cases may involve large tariff differentials between declared and predicted categories. Second, legitimate novel products refer to genuinely new or substantially different items not represented in the training data, often emerging from technological innovation. These products may be flagged as anomalies or misclassified product categories because they do not align with learned representations. Third, concept drift occurs when the statistical properties of product descriptions change over time, such as shifts in terminology or updates to HS rules. As a result, previously well-classified products may appear anomalous or misclassified because the model's learned decision boundaries no longer reflect the

current data distribution. For the latter two categories, the deviations primarily arise from product classification inconsistencies and are not necessarily linked to substantial tariff differentials. Nevertheless, all cases require review and validation by customs authorities to ensure regulatory compliance.

The focus is on identifying cases where products have been assigned to incorrect categories within textual trade data. This validation layer can be integrated into a system that aims to reduce product misclassification rates and improve trade compliance accuracy. The central hypothesis of this study is that combining contextualised text embeddings with autoencoder architectures will improve the identification of contextual anomalies, which is important for detecting product misclassification. Our approach integrates two primary components: pretrained transformer-based language models and autoencoder architectures. Given that autoencoders have various architectures, this experiment proposes several configured autoencoder architectures based on the justification that architectural differences influence detection performance. Finally, the study performs a comparative analysis to determine the optimal network architecture suitable for trade data characteristics in detecting anomalies in semi-supervised and unsupervised learning paradigms.

This thesis advances the field of text-based anomaly detection by addressing the problem of product misclassification in international trade, with the following contributions:

1. We propose a method for text anomaly detection by integrating autoencoder architectures with contextualised embeddings derived from pretrained language models. Three variants of autoencoder with different architectures were developed and evaluated to understand their effectiveness in learning latent representation and to determine the optimal architecture for detecting product misclassification in the trade domain.
2. We propose anomaly detection trained on both semi-supervised and unsupervised learning paradigms using different training samples with varied contamination values and rates of anomalies. This approach contributes to evaluate the model robustness and applicability in the real-world trade classification systems.

This chapter is structured as follows. Section 5.2 presents the proposed model architectures, which detailing the mechanisms of various autoencoder networks and their integration with the text encoder. Section 5.3 describes the dataset, baseline models, evaluation metrics, and experimental settings. Section 5.4 reports the experimental results both in a semi-supervised and unsupervised approach, compares model performance, and discusses the practical advantages of the proposed approach. Finally, Section 5.5 provides a summary of the chapter.

5.2 Methodology

5.2.1 Semi-supervised and Unsupervised Learning with Autoencoder

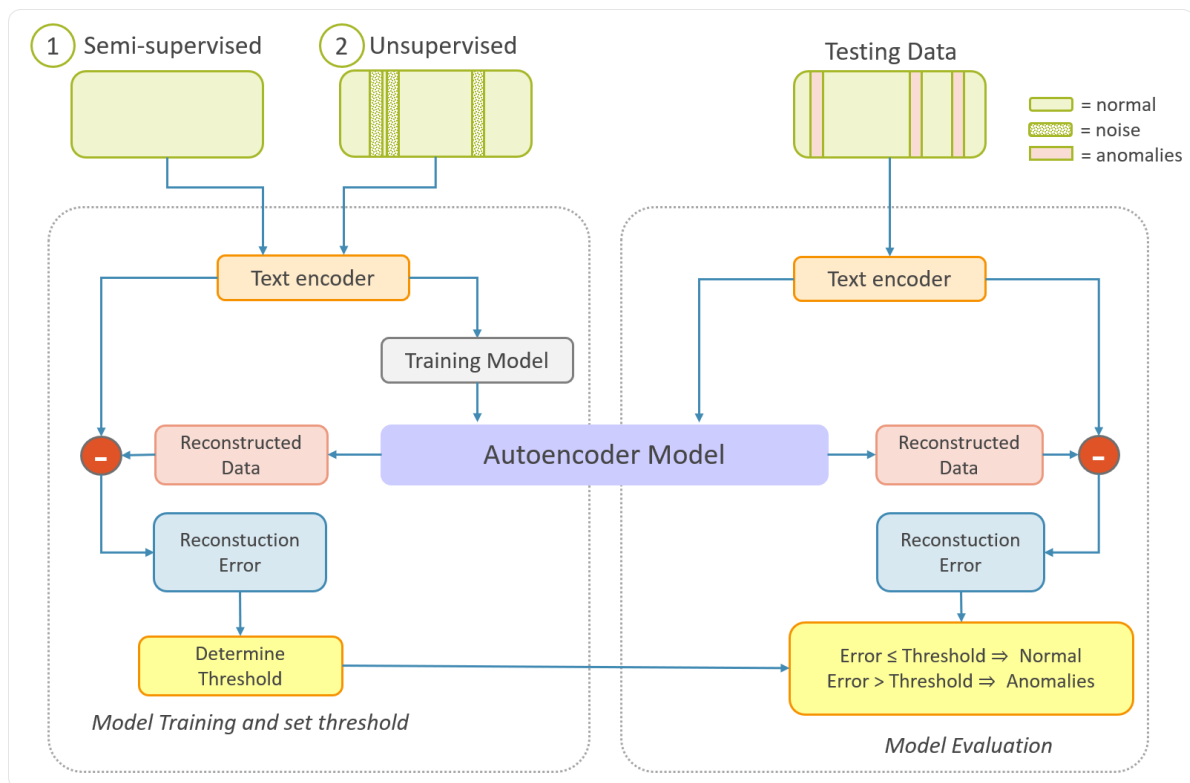


Figure 5.1 A general overview of the proposed text-based anomaly detection framework for identifying misclassified trade transactions. The approach incorporates both semi-supervised and unsupervised learning methods to train the models. Anomalies are subsequently detected using a reconstruction-based mechanism, with threshold determination derived from the reconstruction error distribution.

The overview of the proposed autoencoder-based anomaly detection framework, which implements two distinct learning paradigms for model training, as illustrated in

Figure 5.1. The first learning paradigm used in our approach is semi-supervised learning. In this approach, the training data contains only normal data. This approach is practical in this context because it reduces the dependency on labelled datasets of anomalous or fraudulent transactions, which are typically scarce and challenging to obtain (Bank *et al.*, 2021). One contributing factor causing the anomalies is regulatory changes that impact economic incentives, such as tariff differentials, which can drive trader behaviours that may lead to potential fraud. Using normal data ensures the validity of the data and simultaneously avoids dependence on trade data that may be affected by policy changes, as product categories evolve continuously and fraudsters adapt their strategies by using ambiguous product descriptions that obscure true product attributes. The proposed semi-supervised learning method is based on the premise that models can acquire useful representations from normal data throughout the training process (Chandola *et al.*, 2009). It detects deviations from established normal patterns without having prior knowledge of particular anomalies. Autoencoders has capability to learn latent representations of normal data patterns, making them effective at capturing deviations from expected normal behaviour. The second approach utilises an unsupervised learning paradigm to train the model using unlabelled datasets. In this context, the training data do not necessarily need to be explicitly identified as normal data but instead are composed of both normal data and noise. Although semi-supervised learning is the most appropriate for our system, getting clean normal data is impractical. Real-world normal trade datasets often contain noise, including typographical errors and incomplete attribute information. The benefits of models trained in an unsupervised setting is their ability to adapt to datasets without requiring manual curation of normal data. Unsupervised learning offers benefits for its ability to identify deviations without prior knowledge of abnormal patterns, even in conditions where diverse and unpredictable linguistic patterns exist in the training data (Chandola *et al.*, 2009; Goldstein and Uchida, 2016; Miguel-Diez *et al.*, 2025).

Both training paradigms mentioned above employ reconstruction-based mechanisms to detect anomalies (Berahmand *et al.*, 2024; Hawkins *et al.*, 2002; Japkowicz *et al.*, 1995). Autoencoders learn compressed latent representations of input data and identify anomalies based on reconstruction error, which assumes that normal patterns can be reconstructed with lower error than anomalous ones. However, the

effectiveness of this mechanism depends on the representation. Ruff suggests that reconstruction-based methods are particularly well suited when the data follow some manifold or prototypical structure (Ruff *et al.*, 2021), where in this case, image data works well since it has lower or simpler underlying structures. Embeddings generated from transformer-based models follow manifold structures (Valeriani *et al.*, 2023), where the Transformer architecture transforms from layer to layer and evolves into high-dimensional representations. In this thesis, transformer-based embeddings are generated as input for reconstruction-based methods to examine their effectiveness in detecting anomalies.

Detecting anomalies in trade data, in particular anomalies within the complexity of product descriptions, often involves contextual anomalies that manifest through subtle linguistic deviations rather than obvious textual errors. In other words, it is necessary to comprehend product descriptions for transactions that usually appear semantically valid while being subtly misclassified into incorrect product categories by learning the latent representation. Adoption of modern language models, such as transformer-based models, is suitable in our case for providing contextualised embeddings that capture semantic relationships and syntactic sequential patterns (Devlin *et al.*, 2019; Sanh *et al.*, 2020). These high-quality embeddings facilitate autoencoders to learn meaningful latent features that effectively distinguish between normal and misclassified product descriptions. Another consideration when developing anomaly detection using autoencoders is the variety of available architectures, including standard, variational, and convolutional autoencoders. Different architectures excel at learning latent representations and reconstructing input data. Therefore, it's important to compare these architectures to find the best one for detecting trade anomalies.

The proposed models distinguish anomalies involving two components, namely reconstruction error and thresholding (Chalapathy and Chawla, 2019). Reconstruction error calculates the difference between the original input and reconstructed output, with higher reconstruction errors indicating potential anomalies. A threshold is then set as a boundary to distinguish between normal and anomalous instances, which is determined from this error distribution. This approach offers flexibility to allow the threshold to be adjusted after training to optimise precision–recall trade-offs based on specific requirements (Pang *et al.*, 2021). The anomaly boundary can be calibrated

after model development, offering flexibility for domain-specific contexts in which the anomaly detection system will be deployed (Ruff *et al.*, 2021).

Problem Formulation

Given a trade dataset $D = \{(s_1, y_1), \dots, (s_i, y_i)\}$, each s_i represents the i -th text sample and $y_i \in \{0, 1\}$ denotes normal (0) or anomalous (1) label, the dataset is subsequently split into $D = D_{\text{normal}} \cup D_{\text{anom}}$ with D_{normal} containing normal data samples and D_{anom} containing anomalous data samples. For semi-supervised learning, the training set $D_{\text{train}}^{\text{semi}} \subseteq D_{\text{normal}}$ contains only normal data samples, while the test set $D_{\text{test}}^{\text{semi}}$ includes anomalous data samples with proportions of anomalies at different rates. In unsupervised learning, the training set, $D_{\text{train}}^{\text{unsup}}$, contains both normal data samples and noise to learn of realistic data patterns. The testing phase is carried out on $D_{\text{test}}^{\text{unsup}}$, keeping the same proportion of anomalies as in the semi-supervised approach to confirm comparability. In the training phase, the training split D_{train} was encoded using a pretrained language model into embeddings, where $X = \{x_1, x_2, \dots, x_k\}$ represents the embedding set, with k denoting the size of the training split. Each embedding $x_i \in \mathbb{R}^d$ is generated from the aggregate token representation of the corresponding text sample s_i . The process is continued by feeding each embedding, x_i , into the autoencoder model.

The autoencoder was trained on each sample data, x_i , and then reconstructed it as $\hat{x}_i$. The model utilises a reconstruction-based anomaly detection mechanism, where anomalies are identified by measuring the discrepancy between the original input, x_i , and its reconstruction $\hat{x}_i$. Data points, which have reconstruction errors surpassing a predefined threshold, will be considered anomalies. This thesis utilises three variants of autoencoder architectures to assess their capability to detect text anomalies. Each model is provided with input data to learn latent representations and to reconstruct the original input, for which the model is then evaluated for their effectiveness in detecting anomalies. A detailed explanation of the architecture and configuration of each model is provided below.

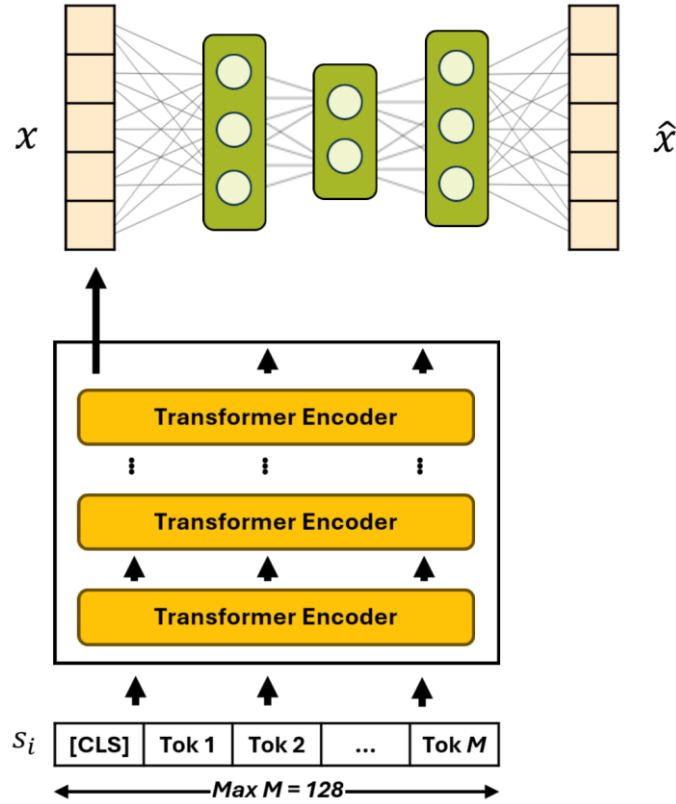
Standard Autoencoder (OAE)

Figure 5.2 Integrated transformer-OAE architecture combining contextual sequence modelling with reconstruction-based anomaly detection.

Our proposed method, standard autoencoder (OAE) was built based on the literature (Bank *et al.*, 2021; Michelucci, 2022), comprises three core components arranged sequentially, the encoder, the latent representation, and the decoder, as depicted in Figure 5.2 The standard or basic autoencoder presented the ability to effectively identify clear deviations from normal patterns, making it suitable as a baseline autoencoder (Sakurada and Yairi, 2014).

The encoder was composed of a series of fully connected layers designed to map the input into a lower-dimensional latent representation, where the encoder was denoted as E and the input was denoted as $x_i \in \mathbb{R}^d$. The latent representation was denoted as $z \in \mathbb{R}^{d'}$, where d' is the compressed dimensionality, such that $d' < d$. The input dimension was set to gradually reduce the dimension from 768 to 512, and then to compress further to an encoded representation of 384 dimensions. The purpose of this design was to avoid information loss and allow the model to learn hierarchical

feature representations. The encoder generated a latent representation denoted as $z = E(x) = \sigma(W_j x + b_j)$. W_j and b_j are the weight matrix and bias vector of the encoder for layer j , and σ represents the activation function, which in this case is ReLU and we employed batch normalisation and dropout for regularisation.

The decoder function, denoted as G , reconstructs the encoded representation and mirrors the encoder structure in reverse to gradually reconstruct back to its original dimensionality. The transformation is denoted as $\hat{x} = G(z) = \sigma(W_k z + b_k)$. W_k and b_k are the weight matrix and bias vector for each layer k of decoder. The decoder used Tanh activation function in the final layer to normalize and produces bounded outputs, $[-1, 1]$. This bottleneck in the architecture encourages the model to learn a latent representation of normal data and functions as an information filter that preserves only important features. The complete network design of the OAE can be seen in Table 5.1.

Component	Input Nodes	Output Nodes	Activation Functions
Encoder	768	512	ReLU
	512	384	ReLU
Decoder	384	512	ReLU
	512	768	Tanh

Table 5.1 OAE network layer specifications contain the encoder that includes two fully connected layers that compress the 768-dimensional input into a 384-dimensional latent representation using ReLU activations. The decoder symmetrically reconstructs the input and uses Tanh in the output layer.

In the training process, the network was optimised by minimising the reconstruction error, which represents the difference between the reconstructed output and the original input data. This error can be quantified using several loss functions such as, Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). Our model utilised MSE, which has been widely adopted for its sensitivity to outliers by penalising larger errors. Formally, the MSE loss is expressed as $\mathcal{L}_{MSE} = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2$, where x_i represents the original input, $\hat{x}_i$ is the reconstructed output, and N is the number of samples.

Variational Autoencoder (VAE)

VAE introduces a probabilistic approach to autoencoders by treating the latent representation as a probability distribution rather than deterministic values (Kingma and Welling, 2014, 2019). Like the standard autoencoder, the model consists of two main components: an encoder and a decoder, with a probabilistic latent space in between, as seen in Figure 5.3. A study performed by Yokkampon (Yokkampon *et al.*, 2020) has shown that VAEs are effective at detecting anomalies specifically applying to time series data, and our purpose in selecting the VAEs in this study is to observe VAEs in learning the underlying probability distribution of text data in capturing anomalies (Berahmand *et al.*, 2024).

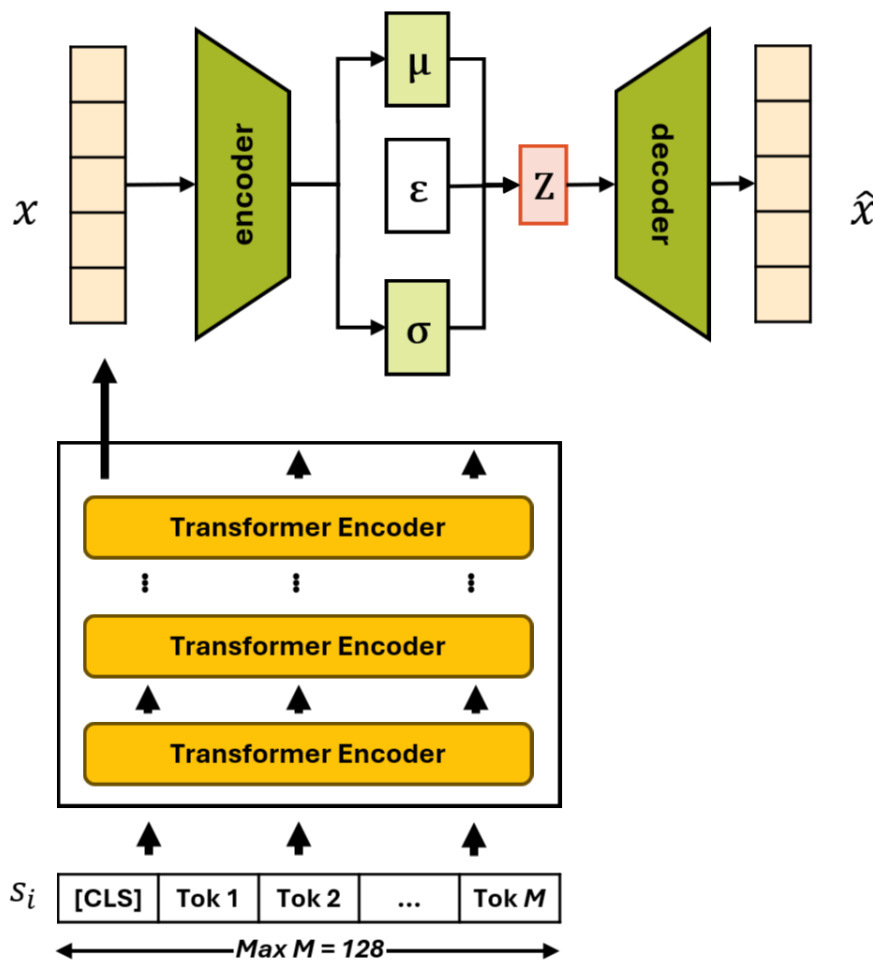


Figure 5.3 Integrated transformer-VAE architecture combining contextual sequence modelling with probabilistic reconstruction for anomaly detection.

The encoder, denoted as E , transforms the input through neural network layers with decreasing dimensions from 768 to 384, where the final layer splits into two outputs:

a mean $\mu = E_\mu(x)$, and log variance $\log \sigma^2 = E_\sigma(x)$, where E_μ and E_σ , are the mean and variance networks, respectively. The μ and $\log \sigma^2$ are used to parameterise the approximate posterior distribution $q(z|x) = N(\mu, \sigma^2)$, with $\sigma^2 = \exp(\log \sigma^2)$. The log variance is predicted instead of variance directly to ensure $\sigma^2 > 0$ without requiring constrained optimisation. To generate latent samples z , the reparameterization trick is employed: $z = \mu + \sigma \cdot \epsilon$, where $\epsilon \sim N(0,1)$ is random noise from a standard normal distribution. The decoder, denoted as G , then takes the sampled latent vector z and reconstructs it back to the original dimensionality through a mirror architecture that expands from 384 to 768 dimensions. The final layer uses Tanh activation to ensure outputs are bounded in $[-1, 1]$, producing the output, $\hat{x} = G(z)$.

The loss function of a VAE consists of two components: a reconstruction loss and a Kullback-Leibler (KL) divergence. The first component measures the loss from data reconstruction, and the latter calculates the divergence term that regularizes the latent space. The KL term encourages the approximate posterior distribution $q(z|x)$ to be close to the prior $p(z)$. The total VAE loss is denoted as a combination of $\mathcal{L}_{VAE} = \mathcal{L}_{MSE} + \beta \cdot \mathcal{L}_{KL}$, where β is a hyperparameter that balances reconstruction reliability against latent space regularisation. For anomaly detection, the reconstruction error is calculated between the input x and its reconstruction, $\hat{x}$. Since the model is only trained on normal data using a supervised learning setting, it is expected that the model has difficulty reconstructing anomalous data. The higher reconstruction error indicates that the input deviates from the normal data distribution, suggesting an anomaly.

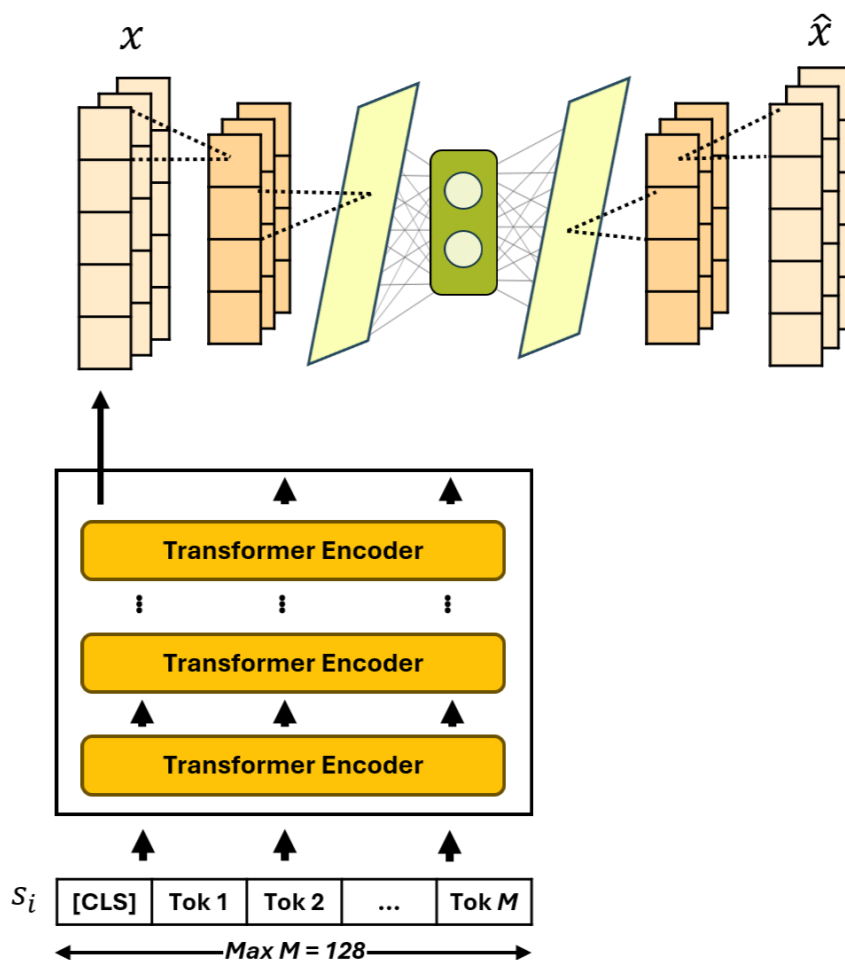
Convolutional Autoencoder (CAE)

Figure 5.4 Integrated transformer-CAE architecture combining sequential modelling with convolutional reconstruction for anomaly detection.

The CNN architecture is commonly utilised in image-based applications, but it has been effectively integrated within autoencoder models (Aamir *et al.*, 2021; Ranjan *et al.*, 2017; Seyfioğlu *et al.*, 2018). The spatial and hierarchical structure inherent in images provides an advantage for using CNN combined with an autoencoder in downstream tasks, such as anomaly detection (Jia and Liu, 2023). In this proposed method, we used CAE to process the text data input, where firstly we utilised, CNN inspired the study of Kim (Yoon Kim, 2014), that has been applied to text data processing, with demonstrating its effectiveness in capturing different levels of abstraction and local dependencies within textual input data. The overview of our proposed CAE model architecture, where the input utilising the contextualised embeddings feeding into the CAE network for anomaly detection, is illustrated in

Figure 5.4. CNN components and processes included in the CAE network for anomaly detection task, such as convolutional layers, pooling layers, concatenation operations, flattening, and FFNs, as illustrated in Figure 5.5.

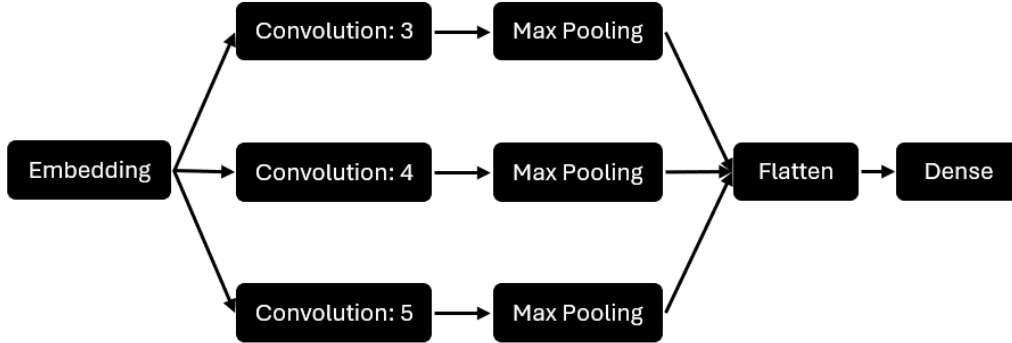


Figure 5.5 Depicts the CAE encoder, where parallel convolutions with kernel sizes 3, 4, and 5 are combined with max pooling, concatenation, flattening, and dense layers to form the feature representation.

The detailed procedure can be formally expressed as follows: given embeddings $x \in \mathbb{R}^{768}$, the encoder E processes the input through m parallel convolutional branches with kernel sizes $K = \{k_1, k_2, \dots, k_m\}$. In this implementation, three parallel branches ($m = 3$), use kernel sizes $k_1 = 3$, $k_2 = 4$, $k_3 = 5$, respectively, a design inspired by (Yoon Kim, 2014). The multiple kernel sizes enable the model to capture patterns at different scales: smaller kernels capture local patterns while larger kernels capture broader contextual patterns. Each parallel branch processes the input independently through convolutional layers, configured with $c_{out} = 150$ output channels per branch. Each convolution is followed by batch normalisation and ReLU activation σ for stable training and non-linearity. Max pooling is applied to each branch to reduce dimensionality while preserving the most prominent features, effectively filtering out noise and emphasising the most important patterns. The output of each branch m is: $h_m = \text{MaxPool}\left(\sigma(\text{Conv1d}(x; k_m, c_{out}))\right)$. The features from all three branches are concatenated and then flattened, which can be formulated as follows: $h_{concat} = [h_1; h_2; \dots; h_m] \in \mathbb{R}^{m \cdot c_{out}}$. The latent representation is computed as $z = E(x) = \sigma(W_E h_{concat} + b_E)$, where $W_E \in \mathbb{R}^{384 \times 450}$ and $b_E \in \mathbb{R}^{384}$. The input dimensionality of 450 results from concatenating three branches of 150 features each ($3 \times 150 = 450$), which is then compressed to 384 dimensions. This bottleneck forces the model to learn the most essential features for reconstruction.

To maintain architectural symmetry and ensure effective reconstruction, the decoder mirrors the encoder design with corresponding parallel branches. Each decoder branch is specifically designed to reconstruct the patterns learned by its corresponding encoder branch. The decoder G reconstructs the original input $x \in \mathbb{R}^{768}$ from latent representation $z \in \mathbb{R}^{384}$, forcing the model to learn compressed representations of normal data patterns. Decoding begins with a fully connected projection: $h' = \sigma(W_G z + b_G)$, expanding the 384-dimensional latent vector to 450 dimensions. The tensor $h' \in \mathbb{R}^{450}$ is then reshaped and split into $(m = 3)$, equal parts: $h'_1, h'_2, h'_3 \in \mathbb{R}^{150}$, corresponding to each decoder branch. Each split h'_i , is processed through a parallel decoder branch: $y_i = \text{AvgPool}\left(\sigma(\text{ConvTranspose1d}(\text{Upsample}(h'_i); k_m, 1))\right)$, where Upsample restores the original sequence length and ConvTranspose1d with kernel size k_m , reduces the 150 channels to 1 output channel. The final reconstruction is obtained by averaging all decoder branch outputs: $\hat{x} = G(z) = \frac{1}{m} \sum_{i=1}^m y_i$, where $(m = 3)$. This ensemble averaging approach consolidates the specialised representations generated by each parallel branch into a unified reconstruction, fully preserving the 768 dimensions of the original embeddings. The model is trained to minimise reconstruction loss $\mathcal{L}_{MSE}$ between the input x and its reconstruction $\hat{x}$, learning to reconstruct normal data patterns.

5.2.2 Anomaly Score and Threshold

In an anomaly detection task, a threshold is an essential component used to establish the decision boundary to differentiate normal from anomalous instances. In this thesis, the threshold is estimated from the reconstruction error distribution of training data. This method has been utilised across diverse datasets in the anomaly detection literature (Berahmand *et al.*, 2024), reducing the need for labelled anomalies in training datasets for anomaly detection (Iqbal and Qureshi, 2023).

A percentile-based threshold was utilised to evaluate anomaly detection performance, which incorporates a contamination parameter, $\alpha \in [0,1]$, that is configured to represent the anticipated proportion of anomalous instances. In this approach, the threshold, τ , is defined as the $(1 - \alpha)$ th percentile of the reconstruction error distribution computed from training samples. This threshold method is intuitive, works with any distribution of data, and adapts to the inherent distributional characteristics of

the reconstruction errors. For illustration, when alpha is set to 0.05, it corresponds to an expected anomaly rate of 5%, and the threshold will be positioned at the 95th percentile of the normal training reconstruction errors. With this configuration, any test sample exhibiting a reconstruction error exceeding this threshold would be classified as anomalous, while lower values than the threshold would be considered normal data. The classification rule is formally defined as:

$$\hat{y}_i = \begin{cases} 1, & \text{if } R_i \leq \tau \text{ (normal)} \\ 0, & \text{if } R_i > \tau \text{ (anomalous)} \end{cases}$$

The reconstruction error is denoted as R_i for sample i , and the predicted class label is defined as $\hat{y}_i$. A domain expert is allowed to calibrate the sensitivity to classify anomalies by adjusting the contamination parameter based on prior knowledge or operational constraints.

5.3 Experiments Settings

5.3.1 Datasets

Xu (Xu, Gábor, *et al.*, 2023) explains data configuration into three strategies for training anomaly detection models for text data. First, generating synthetic anomalous samples gives control over the size of anomalies, but may risk producing anomalies that do not reflect real-world irregularities. A second strategy was performed by combining multiple datasets, where normal data is obtained from one or more corpora, and anomalous samples originate from a different source. However, this approach raises a question about domain mismatch. The third strategy is considered the most widely used approach, where one category within a corpus is specified as anomalies while treating the others as normal (Barrett *et al.*, 2019; Tec Yan Yap, 2020).

This thesis adopted the third dataset configuration strategy, where both normal and anomalous data originate from the same dataset. This arrangement was better to reflect the problem since product misclassification occurred within the same corpus rather than from an entirely different source. The key challenge, therefore, lies in determining which categories represent normal behaviour and which represent anomalies. We set the following criteria to define anomalies. First, there must be a significant gap in duty or tax rates between normal and anomalous categories. Our

dataset is a publicly available Indian trade dataset⁴, previously used in Harsani (Harsani *et al.*, 2020). We associated each product transaction with a specific duty rate obtained from regulatory documents, which provides the basic customs duty rates, as shown in Figure 5.6. The substantial disparities present in the tariff for certain products in product categories, for example, HS code 84713010 is exempted from tariff with 0% rate, but HS code 85182100 is subject to a 10% basic rate.

Tariff Item	Description of goods	Unit	Rate of duty	
			Stand ard	Prefer- ential Areas
(1)	(2)	(3)	(4)	(5)
8471	AUTOMATIC DATA PROCESSING MACHINES AND UNITS THEREOF; MAGNETIC OR OPTICAL READERS, MACHINES FOR TRANSCRIBING DATA ON TO DATA MEDIA IN CODED FORM AND MACHINES FOR PROCESSING SUCH DATA, NOT ELSEWHERE SPECIFIED OR INCLUDED			
8471 30	- <i>Portable automatic data processing machines, weighing not more than 10 kg, consisting of at least a central processing unit, a keyboard and a display:</i>			
8471 30 10	--- Personal computer	u	Free	-
(1)	(2)	(3)	(4)	(5)
	MICROPHONE AND ONE OR MORE LOUDSPEAKERS; AUDIO-FREQUENCY ELECTRIC AMPLIFIERS; ELECTRIC SOUND AMPLIFIER SETS			
8518 10 00	- Microphones and stands therefor	u	10%	-
	- <i>Loudspeakers, whether or not mounted in their enclosures :</i>			
8518 21 00	-- Single loudspeakers, mounted in their enclosures	u	10%	-

Figure 5.6 Comparison tariff classifications and duty rates for selected products, including HS 84713010 (computers) and HS 85182100 (audio equipment). It presents that computers incur no import duty, while audio equipment is taxed at 10%.

Another key consideration in selecting the anomalous category was the identification of product categories known to present classification challenges because of their variation and high trading volume (Fedotova, 2021). Machinery and mechanical devices, classified under HS Chapters 84 and 85, have been recognized in the literature as frequently subject to misclassification in international trade (de Lima *et*

⁴ <https://www.zauba.com>

al., 2022). These chapters include product categories, which present ambiguity in manual verification processes. These product categories are suitable for a contextual anomaly detection case because ambiguity frequently arises from the use of overlapping or vague terminology in product descriptions. For instance, some terms in product descriptions, like workstation, PC and computer, are frequently used to refer to similar products, causing inconsistencies in classification. Examples of such transactions are presented in Table 5.2.

HS code	Rate	Example	Label
84713010	0% (Free)	<i>appc0387 z0rg apple macbook pro 15 with retina -mdl:a1398 bis regno:r-41000450 dt:30.08.2013 eta no:nr-eta-4123/24 dt:1</i>	Normal
85285900	10%	<i>electronic goods - continuous workstation monitor part no: 023 007, model :wsm-02 serial no: 11189 with interconnection</i>	Anomaly
85183000	10%	<i>calisto p825m speaker phone for pc & mobile phone w/built in mic 84261</i>	Anomaly
84145990	12.5%	<i>fan tray (012-0439-029-93) part of fire control computer assy, sl no:18 p/n:d0400-1100-501 (bel p/n:476915640148) boe no</i>	Anomaly

Table 5.2 Examples of the transactions in the datasets. Products with a 0% (duty-free) rate are labelled as normal. In contrast, products with higher duty rates are treated as anomalous in our experiment, as they will be deliberately assigned the same HS code used for normal data.

Both learning paradigms were implemented in these experiments to train the models using semi-supervised and unsupervised learning. In the semi-supervised setting, the training data contained only normal data samples, while the testing data included anomalies at three distinct rates of anomaly levels at 2%, 5%, and 8% of the test data split. The variation in anomaly rate functioned to evaluate the model's robustness in detecting anomalies under different levels of anomalous rates, simulating realistic uncertainty scenarios. Additionally, the models trained under unsupervised learning, 20% of the normal training data was deliberately perturbed with synthetic noise. These noisy data samples are synthetic perturbations created by modifying normal data samples to challenge model robustness to detect anomalies. Data perturbation

adopted a multi-stage approach by selecting normal training data randomly (Bayer *et al.*, 2023; Li, Hou, *et al.*, 2022). Text preprocessing involved the removal of non-alphabetic characters and the removal of numeric tokens. Then, the selected samples are evenly divided among several types of noise: character-level perturbations (adjacent character swapping, random character insertion, and character deletion) and token-level perturbations (token deletion), as illustrated in Table 5.3. This approach created a controllable synthetic variation to evaluate model robustness against typographical errors and missing information usually presented in the trade data. This approach was inspired by prior work employing noisy samples in the normal data (Dai *et al.*, 2024) and was further justified as a means to balance the impact of contamination on model performance relative to realistic conditions (Masakuna *et al.*, 2024). The testing data for unsupervised and semi-supervised learning had a similar configuration. Text preprocessing included the process of converting product descriptions to lowercase and removing punctuation.

Applied Noise Process	Noisy Text Examples
Original	<i>appc0387 z0rg apple macbook pro 15 with retina -mdl:a1398 bis regno:r-41000450 dt:30.08.2013 eta no:nr-eta-4123/24 dt:1</i>
Noise:	
• Cleaning alphanumeric	<i>appc zrg apple macbook pro with retina mdla bis regnor eta nonreta</i>
• Swapping adjacent characters	<i>appc rzg apple macbook pro iwth retian mdla bis regnor tea nonreta</i>
• Inserting random characters	<i>appc rpzg apple macbook pro iwwth retian mdla bis rxegnor tjea nonreta</i>
• Deleting random characters	<i>apc rpz aple macbook pro iwwh retian mdla bis rxegnor tjea nonreta</i>
• Deleting random tokens	<i>apc macbook mdla bis rxegnor nonreta</i>

Table 5.3 Examples of noise steps treatment to a product description.

Learning	Samples	Training Data	Testing Data
<i>Semi-supervised:</i>			
Normal	27,648	24,883	2,765
Anomalies	4,549	-	2%, 5%, 8%
<i>Unsupervised:</i>			
Normal	27,648	19,907 + 4,976 (noise)	2,765
Anomalies	4,549	-	2%, 5%, 8%

Table 5.4 Distribution of normal and anomalous samples in the dataset. Anomalous samples in the testing set are a variant proportion as a percentage of the normal data in the testing split.

5.3.2 Baseline Models

The experiments include baseline models for comparative evaluation with the proposed methods. These traditional methods provide justification for using neural-network models using autoencoders.

- OC-SVM is a boundary-based approach that uses kernel functions to map data into high-dimensional space and finds a hyperplane that separates normal training data from the origin while maximising the margin. Unseen data are classified based on the decision score relative to the threshold to distinguish normal and anomalous data. The proposed method will be compared with OC-SVM, which has been widely adopted as a baseline for anomaly detection methods in various studies (Hejazi and Singh, 2013; Yang *et al.*, 2021; Zhang *et al.*, 2015).
- Isolation Forest (IFO) (Liu *et al.*, 2008) is a tree-based ensemble method for anomaly detection that works by recursively partitioning the data. The idea behind this algorithm is based on the assumption that anomalies are relatively rare and often distinct, so they tend to be isolated after fewer splits, resulting in shorter average path lengths that manifest as the anomaly score. In addition to its efficiency, the algorithm has also been widely used as a benchmark in anomaly detection studies (Ding and Fei, 2013; Xu, Pang, *et al.*, 2023).
- Local Outlier Factor (LOF) (Breunig *et al.*, 2000) is a density-based method that identifies anomalies by comparing the local density of a data point relative to its

neighbours. This algorithm estimates the local reachability density using reachability distances to the k -nearest neighbours, in this case, k is a user-defined parameter that determines the size of the local neighbourhood for density estimation. Data points with density similar to their neighbours receive LOF scores close to 1.0 (normal), while data points with substantially lower density yield higher LOF scores (potential anomalies).

All baseline models were provided with input generated from the pretrained language models, which was used similarly for training the autoencoder networks to ensure a fair basis of comparison across algorithms. The baseline models also required the specification of an expected anomaly rate, which is generally parameterised through contamination parameters. The parameters indicate the assumed proportion of anomalies within the dataset. To illustrate, setting the contamination rate at 0.05 instructs the algorithm to expect approximately 5% of instances to be anomalous. The contamination values were set to assess parameter sensitivity with different values at 0.01, 0.05, 0.10, and 0.15. These values were then examined for their impact on model performance when the testing data contained anomalies with different rates ranging from 2%, 5% and 8%.

5.3.3 Evaluation Metrics

The model performance utilised the Area Under the Curve (AUC) score, a metric commonly used for comparing and ranking classifiers for anomaly detection (Bradley, 1997; Hanley and McNeil, 1982; Ruff *et al.*, 2021). Additionally, Receiver Operating Characteristic (ROC) curves are utilised because they help to identify suitable decision thresholds and offer more insight into model behaviour than a single scalar value. The model evaluation was carried out using a stratified 10-fold CV to compare the baseline model with the proposed method, and semi-supervised versus unsupervised learning approaches.

5.3.4 Experiments Settings

Building on the effectiveness of language models for trade data shown in Chapter 3, this study adopted embeddings from the transformer-based DistilBERT (base-uncased) model (Sanh *et al.*, 2020). DistilBERT provides a balance between strong classification performance and computational efficiency using trade datasets. Besides

contextualised embeddings, we incorporate alternative text representation methods for comparison. These include traditional approaches such as TF-IDF (Salton and Buckley, 1988) and non-contextualised embeddings including Word2Vec (Mikolov *et al.*, 2013), Glove (Pennington *et al.*, 2014) and FastText (Bojanowski *et al.*, 2017). The use of diverse text representations serves two purposes. First, benchmarking the model performance against the transformer-based approach, and examining the capacity of the autoencoder network to handle different types of text representation.

All experiments and data analysis procedures were conducted using P100 GPUs with 16 GB of RAM on the high-performance computing facilities at Cardiff University, managed by Advanced Research Computing at Cardiff (ARCCA). The default hyperparameters for the model configuration are presented in Table 5.5.

Hyperparameter	Value	Hyperparameter	Value
Embedding dimension	768	Learning rate	2×10^{-5}
Pre-trained LM	distilbert-based-uncased	Latent dimension	384
Max length tokens	128	Dropout	0.2
Epoch	5	Optimizer	Adam
Batch size	16	Contamination/nu	0.01, 0.05, 0.10, 0.15

Table 5.5 Experimental hyperparameter configuration.

5.4 Result and Discussion

The initial part of this section presents the empirical results of our proposed method in identifying misclassified data in a semi-supervised learning framework. Model evaluation focused on the model performance, examining key components that contribute to the model's effectiveness and continued to evaluate the unsupervised learning approach. Lastly, the section was concluded with a model comparison between semi-supervised and unsupervised performance.

5.4.1 Semi-Supervised Approach

Model Performance

The experiment evaluated the proposed model's performance using semi-supervised

learning settings with AUC as the primary metric. Experiment results categorised into traditional baseline methods and autoencoder approaches, evaluated across varying anomaly levels in the test data and different contamination parameter settings during model training, as presented in Table 5.6. The CAE was the strongest competitor to the OAE, with the VAE ranking next. In traditional methods, LOF achieved the highest AUC scores above 93% across all contamination levels. In contrast, both OC-SVM and IFO exhibited lower performance, with OC-SVM presenting fluctuating AUC scores across different anomaly rates. In general, the OAE achieved the best performance among the evaluated models, presenting significant improvements across all evaluation metrics, as shown in Table 5.7. Wilcoxon tests revealed statistically significant differences in favour of the OAE over all baseline methods, with large effect sizes underlining strong practical relevance.

Our finding corresponds with Li's study (Li *et al.*, 2021), which demonstrated the capability of autoencoder-based models to detect text anomalies. However, our approach extended this study by leveraging contextualised embeddings from pretrained transformer-based language models combined with autoencoder architectures, instead of using GRU based language model for text representations. The comparison of OAE with other autoencoder variants revealed substantial effect sizes, underlining that a simple architectural design is suitable to achieve performance improvements in text-based anomaly detection.

Cont./nu		0.01			0.05		
Anom. Rate	2%	5%	8%	2%	5%	8%	
<i>Baseline:</i>							
OC-SVM	0.819±0.036	0.943±0.020	0.966±0.012	0.972±0.017	0.824±0.011	0.947±0.008	
IFO	0.916±0.033	0.916±0.033	0.916±0.033	0.910±0.015	0.920±0.027	0.920±0.027	
LOF	0.934±0.011	0.934±0.011	0.934±0.011	0.932±0.007	0.931±0.005	0.931±0.005	
<i>Proposed:</i>							
OAE	<u>0.993±0.001</u>	<u>0.993±0.001</u>	<u>0.993±0.001</u>	<u>0.993±0.002</u>	<u>0.993±0.001</u>	<u>0.993±0.001</u>	
VAE	0.952±0.013	0.950±0.018	0.948±0.018	0.950±0.039	0.948±0.009	0.950±0.011	
CAE	0.988±0.003	0.988±0.003	0.988±0.003	0.987±0.006	0.988±0.002	0.988±0.002	

Cont./nu		0.10			0.15		
Anom. Rate	2%	5%	8%	2%	5%	8%	
<i>Baseline:</i>							
OC-SVM	0.966±0.012	0.968±0.006	0.968±0.005	0.974±0.008	0.975±0.005	0.975±0.004	
IFO	0.916±0.033	0.920±0.027	0.921±0.027	0.916±0.033	0.920±0.027	0.921±0.027	
LOF	0.934±0.011	0.931±0.005	0.931±0.004	0.934±0.011	0.931±0.005	0.931±0.004	
<i>Proposed:</i>							
OAE	<u>0.993±0.001</u>	<u>0.993±0.001</u>	<u>0.993±0.001</u>	<u>0.993±0.001</u>	<u>0.993±0.001</u>	<u>0.993±0.001</u>	
VAE	0.948±0.018	0.950±0.013	0.952±0.007	0.952±0.018	0.951±0.008	0.951±0.008	
CAE	0.988±0.003	0.988±0.002	0.988±0.002	0.988±0.003	0.988±0.002	0.988±0.002	

Table 5.6 presents the AUC performance of baseline models versus the proposed autoencoder-based models, the models were tested under various anomaly rates and contamination parameters using semi-supervised learning with 10-fold CV.

Models	<i>p</i>-value	Wilcoxon	Mean Difference	Effect Size (Cohen's <i>d</i>)
<i>Baseline:</i>				
OC-SVM	<0.001	<0.001	0.0659	1.4578
IFO	<0.001	<0.001	0.0737	3.7448
LOF	<0.001	<0.001	0.0612	11.8245
<i>Proposed:</i>				
VAE	<0.001	<0.001	0.0423	5.1097
CAE	<0.001	<0.001	0.0049	2.8688

Table 5.7 Statistical comparison of the OAE against baseline models and other autoencoder variants in semi-supervised learning settings.

ROC curves are reported to support visual analysis in capturing threshold-dependent performance characteristics to complete scalar metrics. Each ROC curve depicted how each model balances between True Positive Rate and False Positive Rate across all classification thresholds, as shown in Figure 5.7. The OAE and CAE presented a steep slope toward the upper-left corner of the plot, underlining high sensitivity and specificity across all anomaly and contamination levels. In contrast, baseline models were positioned among the lower-performing models. Additionally, the VAE surprisingly ranks lower among the autoencoder variants in our study, contrasting with findings from Rybak (Rybak and Hassall, 2021), who specifically selected VAE for its advanced statistical tools over standard autoencoders in anomaly detection tasks.

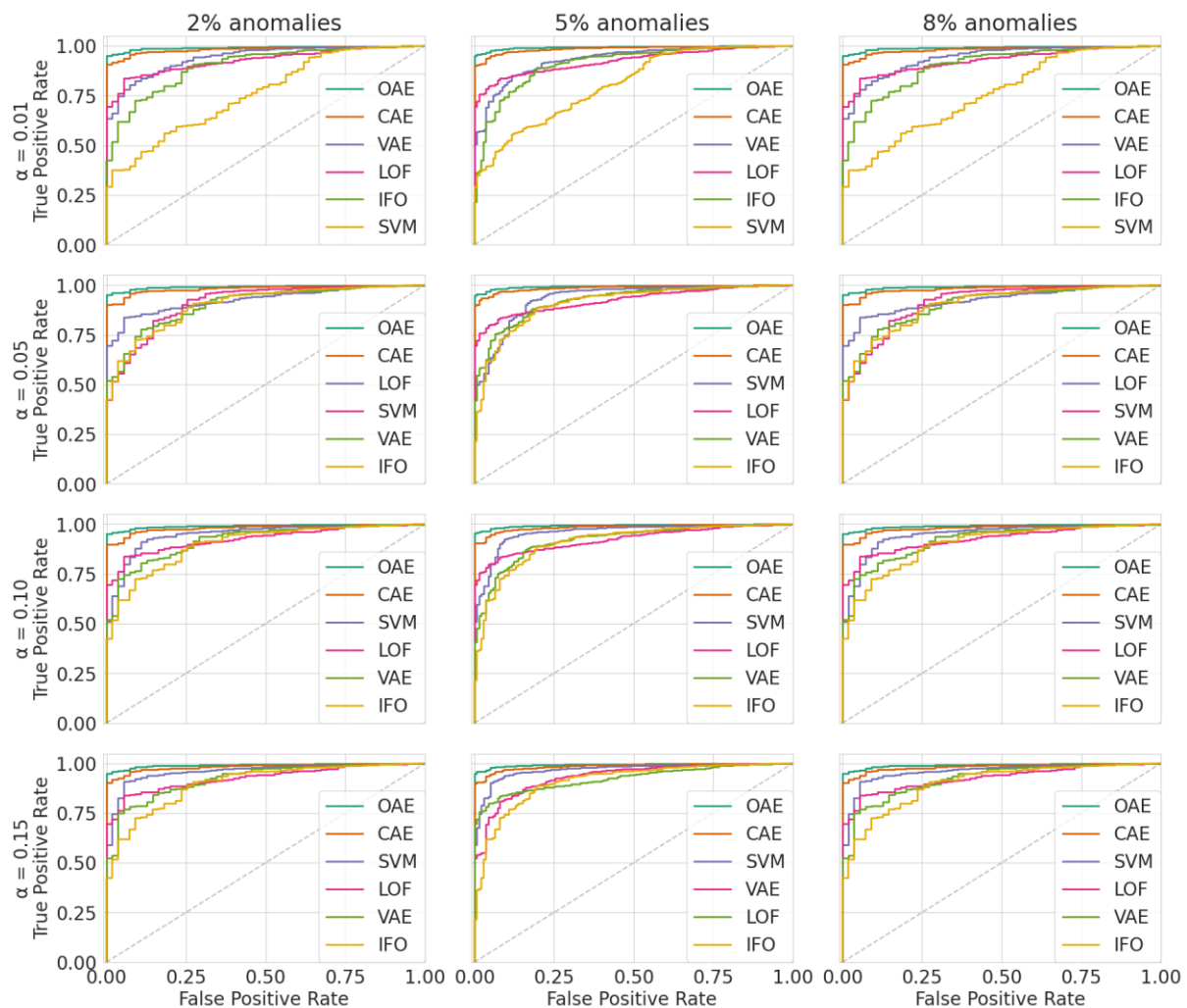


Figure 5.7 Presents the ROC curve comparison of anomaly detection algorithms under varying contamination rates and anomaly levels. The OAE consistently outperformed all baseline methods and other autoencoder variants.

Latent Representation

To better understand the anomaly detection mechanism, visualisations of the embeddings and the corresponding latent representations generated by the autoencoder are presented. PCA was utilised for dimensionality reduction, enabling a visual analysis of clustering patterns and the separation between normal and anomalous instances.

The embedding visualisation of the testing data depicts the distribution of normal and anomalous samples, as shown on the left side of Figure 5.8. The overlap between normal and anomalous data indicated the complexity of the anomaly detection task to separate these categories. The overlapping data points underline the obstacle in

determining decision boundaries in high-dimensional text data, where semantic similarities in normal data and anomalies present to add complexity.

The latent space representation demonstrates how the autoencoder's learned compression affects the geometric separation between normal and anomalous text samples, as shown in the right side of the column of Figure 5.8. The OAE presented improvement in class separation, with its standard feed-forward architecture forming the anomalies located in distinct region clusters of the latent space.

The latent representation of the VAE presents smooth and compressed clusters where normal and anomalous samples often overlap, as seen in Figure 5.8. VAEs used a unimodal Gaussian prior, which struggles to model multimodal distributions, forcing the encoder to map diverse text patterns toward one central region, collapsing natural distributional differences into overlapping regions. This may be the reason why the VAEs are unsuitable for anomaly detection tasks, where preserving distinct clusters between normal and anomalous patterns is crucial.

On the other hand, a dramatic transformation was observed in the latent representation generated by the CAE, which appears to achieve optimal linear separability in the latent space. The CAE effectively learned the latent representation of the data using its convolutional layers, capturing spatial and structural features allowing clearer separation between normal and anomalous data. The CAE generated latent embeddings in a large, connected manifold for normal data, and anomalous samples cluster distinctly in separate regions, each reflecting different types of structural difference, as shown in Figure 5.8 (bottom).

The experimental result of examining latent representation provides empirical evidence that each autoencoder variant transforms the geometric structure of the latent space. Additionally, the visualisation also supports understanding the geometric structure, supplementing quantitative metrics. This approach accelerated intuitive assessment of the architectural design that influenced the transformation of input data into latent representations, thus supporting interpretation and understanding of model behaviour.

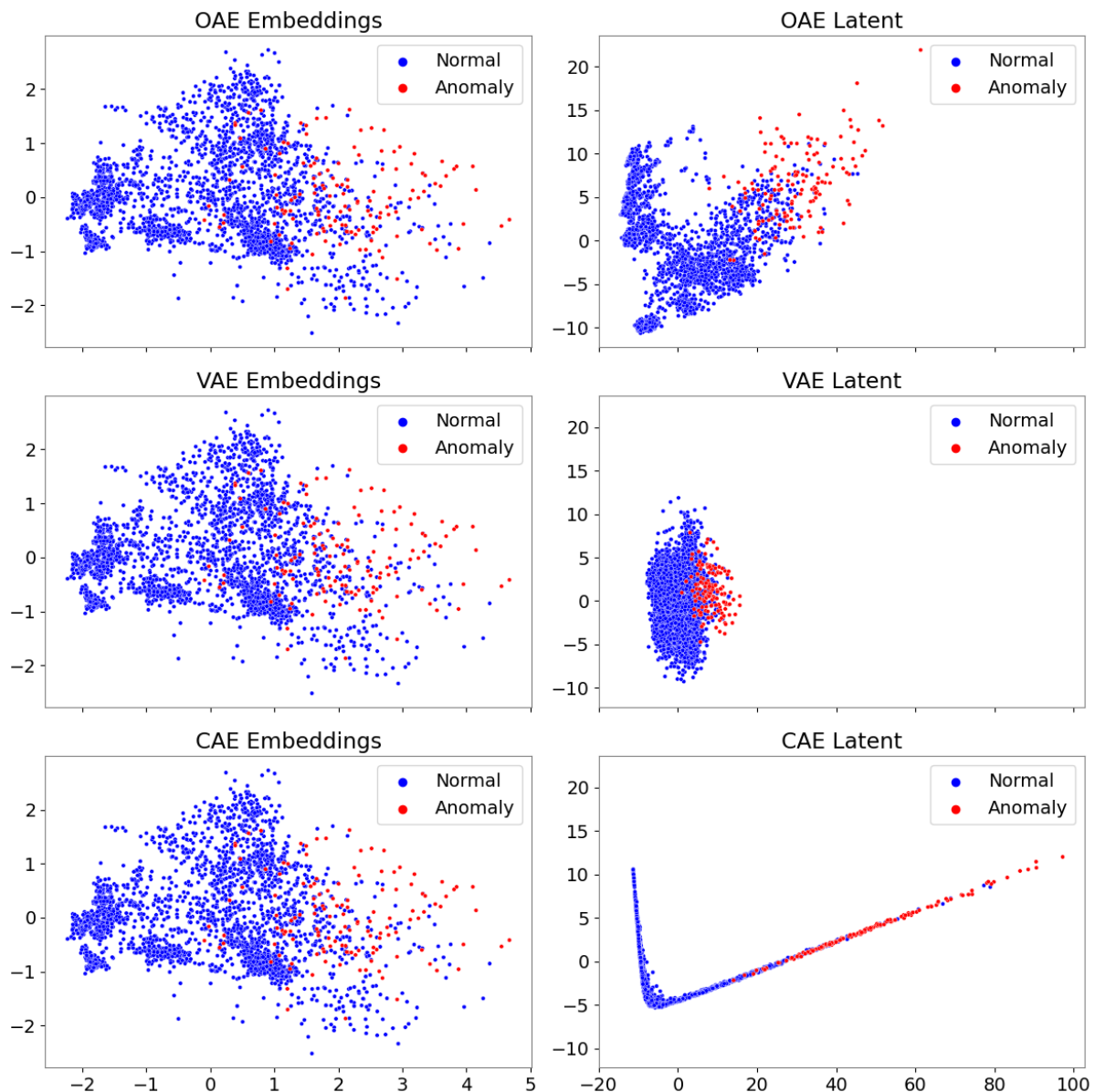


Figure 5.8 Visualisation of test data embeddings and corresponding latent representations using PCA for OAE, VAE, and CAE models (5% anomaly rate, contamination/nu = 0.05).

Hidden Layers

The analysis of hidden layer configurations aims to understand autoencoders' learning feature representations and capture textual patterns related to anomaly detection. The analysis of the hidden layer of autoencoders is necessary for our proposed method using text data, where anomalies may manifest at multiple linguistic levels, requiring different degrees of representational abstraction.

From the theoretical perspective, the depth of the network influences the learning

capacity, where the shallow hidden layers capture basic patterns and deeper layers learning more complex representations (Bengio *et al.*, 2006; Goodfellow *et al.*, 2016). However, our empirical findings revealed that feeding high-quality contextualised embeddings to shallow autoencoders [768, 384] performed comparably to deep autoencoders [768, 384, 192, 96, 48, 24]. These results showed no statistically significant performance difference, as seen in Table 5.8. These results show that contextualised word embeddings sufficiently captured the semantic complexity of text data, empowering autoencoder architectures to learn the latent representation to construct the original input.

Cont./nu	0.01			0.05		
Anom. Rate	2%	5%	8%	2%	5%	8%
<i>Hidden Layer:</i>						
Deep	0.993±0.001	0.993±0.001	0.993±0.001	0.993±0.001	0.993±0.001	0.993±0.001
Shallow	0.993±0.001	0.993±0.001	0.993±0.001	0.993±0.001	0.993±0.001	0.993±0.001

Cont./nu	0.10			0.15		
Anom. Rate	2%	5%	8%	2%	5%	8%
<i>Hidden Layer:</i>						
Deep	0.993±0.001	0.993±0.001	0.993±0.001	0.993±0.001	0.993±0.001	0.993±0.001
Shallow	0.993±0.001	0.993±0.001	0.993±0.001	0.993±0.001	0.993±0.001	0.993±0.001

Table 5.8 Presents the AUC performance of shallow and deep autoencoder layers models, the models were tested under various anomaly rates and contamination parameters using semi-supervised learning with 10-fold CV.

Text Feature Representation

The comparative evaluation of text representation techniques, including traditional methods (TF-IDF) and non-contextualised embeddings (Word2Vec, GloVe, FastText), serves as baseline validation to systematically assess the effectiveness of different text representation methods for anomaly detection in textual data. This analysis was performed to examine our assumption that contextualised embeddings offer a benefit for autoencoder-based detection of contextualised anomalies in text-based anomaly detection research. Different generated text representation was observed to capture different linguistic properties that may affect anomaly detection performance fundamentally.

As seen in Table 5.9, it is evident that integrating transformer-based embeddings with OAE outperformed other text representation methods when combined with the same autoencoder architecture. Among non-contextual text representations, GloVe achieved the best performance, followed by Word2Vec and FastText, while TF-IDF had the lowest performance. The OAE performance exhibited statistically significant performance differences when using transformer-based embeddings (DistilBERT), as it suggests better capture of the underlying semantic and syntactic patterns of normal text, allowing for accurate identification of anomalous instances, as shown in Table 5.10. The large effect sizes further emphasized the practical significance of these results, justifying the computational overhead associated with transformer-based representations.

Cont./nu	0.01			0.05		
Anom. Rate	2%	5%	8%	2%	5%	8%
<i>Language Model:</i>						
TF-IDF	0.560±0.028	0.555±0.016	0.551±0.008	0.660±0.034	0.652±0.015	0.645±0.015
Word2Vec	0.774±0.036	0.788±0.028	0.788±0.021	0.942±0.018	0.949±0.009	0.947±0.007
GloVe	0.813±0.041	0.828±0.029	0.828±0.015	0.955±0.013	0.960±0.008	0.962±0.005
FastText	0.642±0.037	0.667±0.031	0.663±0.021	0.860±0.022	0.873±0.016	0.875±0.009
OAE (DistilBERT)	<u>0.993±0.001</u>	<u>0.993±0.001</u>	<u>0.993±0.001</u>	<u>0.993±0.002</u>	<u>0.993±0.001</u>	<u>0.993±0.001</u>

Cont./nu	0.10			0.15		
Anom. Rate	2%	5%	8%	2%	5%	8%
<i>Language Model:</i>						
TF-IDF	0.711±0.033	0.698±0.015	0.696±0.015	0.742±0.043	0.723±0.021	0.736±0.028
Word2Vec	0.950±0.004	0.950±0.004	0.950±0.004	0.925±0.005	0.925±0.005	0.925±0.005
GloVe	0.949±0.005	0.950±0.004	0.950±0.004	0.926±0.006	0.926±0.006	0.926±0.005
FastText	0.923±0.012	0.926±0.006	0.924±0.007	0.922±0.007	0.921±0.007	0.923±0.006
OAE (DistilBERT)	<u>0.993±0.001</u>	<u>0.993±0.001</u>	<u>0.993±0.001</u>	<u>0.993±0.001</u>	<u>0.993±0.001</u>	<u>0.993±0.001</u>

Table 5.9 AUC performance comparison of various text representation methods across different anomaly rates and contamination parameters (nu), evaluated under a semi-supervised learning with 10-fold CV.

Language Model	p-value	Wilcoxon	Mean diff.	Effect Size (Cohen's d)
TF-IDF	< 0.001	< 0.001	0.33	6.54
Word2Vec	< 0.001	< 0.001	0.09	1.83
GloVe	< 0.001	< 0.001	0.08	1.96
FastText	< 0.001	< 0.001	0.15	1.90

Table 5.10 Statistical comparison of OAE with transformer embeddings versus alternative text representations in semi-supervised learning, where all comparisons favour to the transformer-based text representation.

Error Analysis

Reconstruction error acts as both a classification metric and a diagnostic tool to assess model behaviour and validate whether the underlying structure of textual patterns has been effectively learned. Figure 5.9 and Figure 5.10 visualise the reconstruction errors for testing samples generated by the OAE and CAE as the two best model performers across different anomaly rates. The results clearly show that normal data exhibit consistently lower reconstruction errors compared to anomalous instances, which display higher values. This visualisation demonstrates both models' capacity to distinguish anomalies based on reconstruction error, with the results indicating that the OAE generally exhibits lower reconstruction error values compared to the CAE.

Additionally, the distribution of reconstruction errors reveals a clear separation between normal and anomalous classes, although some overlap remains, as presented in Figure 5.9 and Figure 5.10. The overlapped region between the normal and anomalous data points represented a region of classification ambiguity, where reconstruction loss struggled to distinguish between the two classes. As a consequence, any selected threshold will inevitably result in misclassifications, showing as either false positives or false negatives.

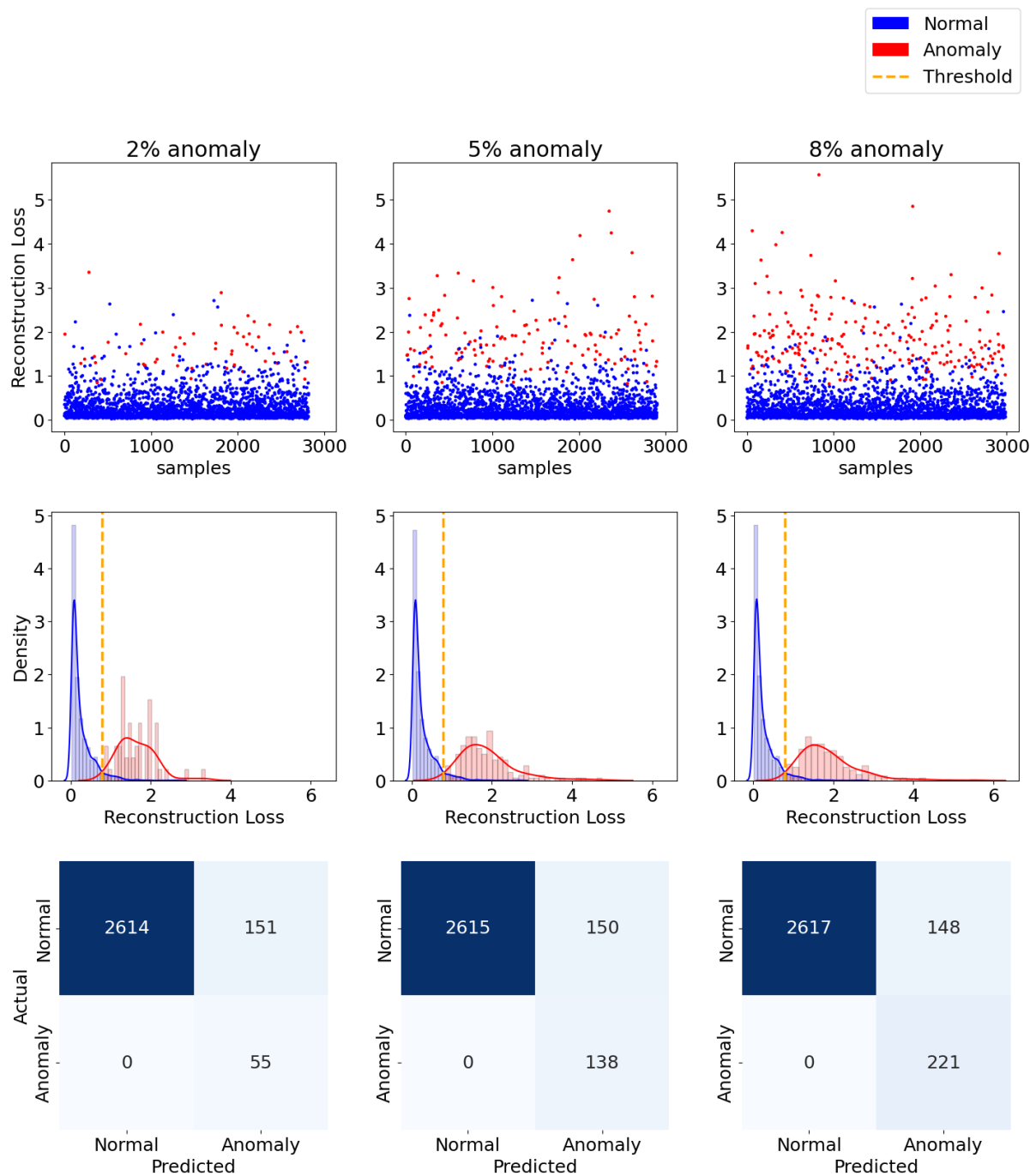


Figure 5.9 OAE performance under semi-supervised learning (5% of anomaly rate and 0.05 of contamination/nu) presenting in scatter plots, loss densities, and confusion matrices.

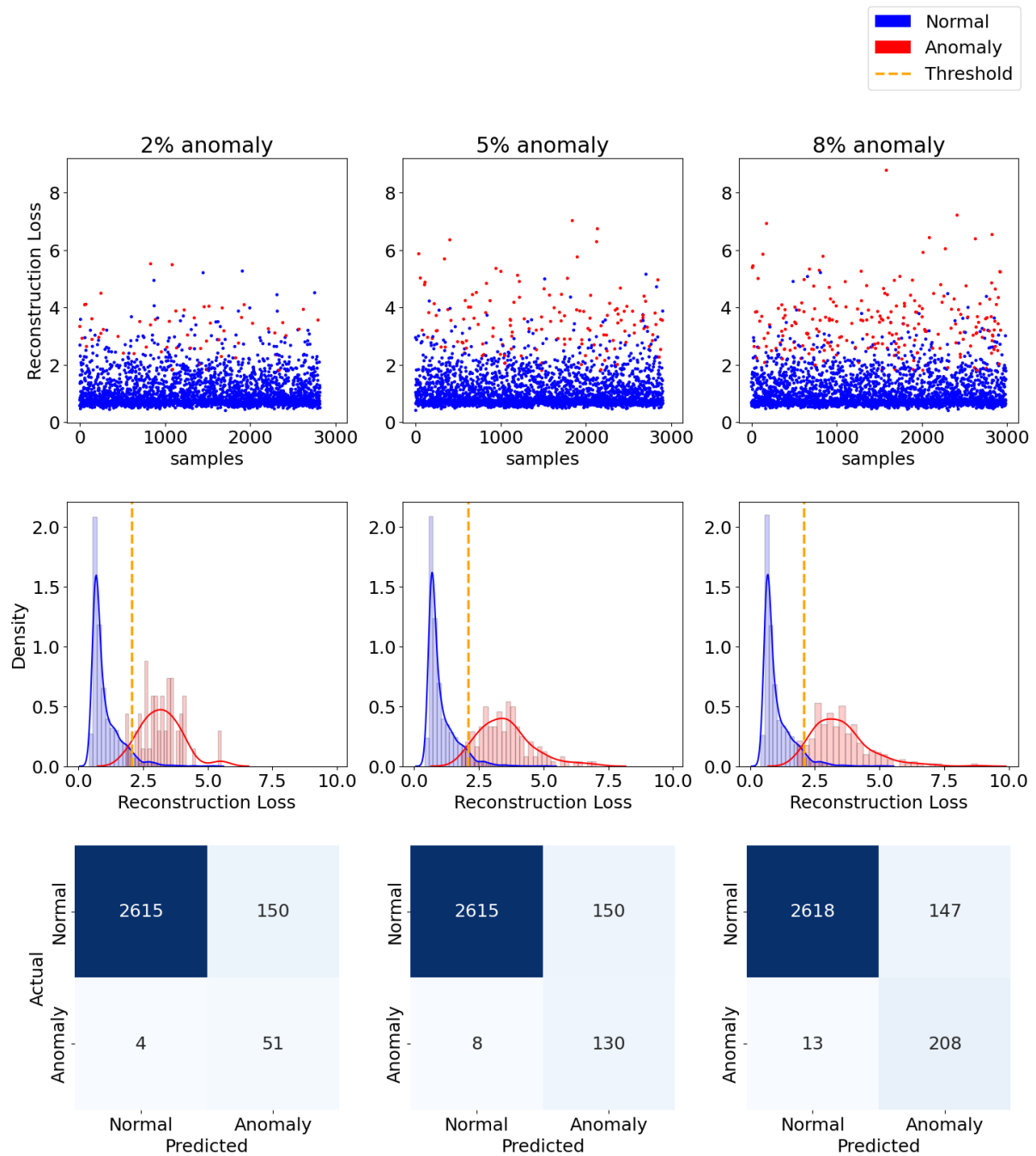


Figure 5.10 CAE performance under semi-supervised learning (5% anomaly rate, contamination/nu = 0.05). The CAE generated greater reconstruction errors than the OAE, which resulted in false negatives and reduced anomaly sensitivity.

To evaluate the anomaly detection model's ability to distinguish between normal and anomalous instances, confusion matrices are presented in Figure 5.9 and Figure 5.10 in the bottom portion of each figure. The results present that the OAE produced no false negative predictions, while the CAE exhibited a progressive increase in false negatives that correlates with higher anomaly rates. An example of a false negative

was *"fan - sps-fan assy gen cpu dx2250 (warranty replacement not for sale) / 438741-001"*, which produced a reconstruction error below the threshold with a prediction result as normal data. The presence of technical terms such as *"cpu"* may have assisted the model in generating lower reconstruction errors. Conversely, several false positives were indicated in the matrix, including examples such as *"laptop (for family personal use)"* and *"computer (dvrs)"*. It seems that the model exhibited higher reconstruction errors when product descriptions included unrelated product terms (e.g., *"office"* or *"family"*) or missing expected brand identifiers (e.g., *"dell"* or *"lenovo"*).

5.4.2 Unsupervised Approach

The unsupervised learning experiment's results presented in this section demonstrate mixed results for each anomaly detection method compared when training using a semi-supervised learning paradigm.

The research results showed that traditional anomaly detection was considerably influenced by the noise present in the training data, resulting in performance degradation with the AUC scores below 90%, as presented in Table 5.12. The results also showed that the OC-SVM model performance was sensitive to the contamination or nu parameter in the unsupervised learning setting, with the AUC scores ranging between 50% and 85%. In contrast, all autoencoder models were less affected by noise or anomalies in the training data and consistently achieved strong results across all scenarios, indicating the robustness of autoencoder models. As presented in Table 5.12, the OAE demonstrates statistically significant performance compared to other models.

Baseline Model	p-value	Wilcoxon	Mean diff.	Effect Size (Cohen's d)
OC-SVM	< 0.001	< 0.001	0.31	5.09
IFO	< 0.001	< 0.001	0.11	5.03
LOF	< 0.001	< 0.001	0.07	10.82
VAE	< 0.001	< 0.001	0.09	6.28
CAE	< 0.001	< 0.001	0.01	3.95

Table 5.11 Statistical comparison of the OAE against baseline models and other autoencoder variants in unsupervised learning settings.

The comparison of model performance reveals that all models trained using semi-supervised learning present statistically significant differences compared to their unsupervised counterparts, as shown in Table 5.13. The OAE achieved the best overall performance when trained under the semi-supervised paradigm. However, semi-supervised training required time-consuming data curation efforts to identify normal training samples. On the other hand, the unsupervised approach provided practical advantages by addressing key limitations of semi-supervised methods, which can be resistant to novel anomaly types and may be proficient in capturing imperfect normal text data present in real-world data.

Beyond differences in training paradigms, long-term deployment introduces additional challenges that extend beyond model architecture and learning strategies. Concept drift poses a challenge for anomaly detection in real-world deployment. As international trade practices, product types, and HS classification standards evolve, the relationship between product descriptions and HS codes may shift. Under such conditions, anomalies may increase because the model relies on outdated representations, leading to gradual performance degradation. Therefore, anomaly detection should be integrated with temporal monitoring and periodic retraining using updated data, aligned with the five-year HS code revision cycle, to ensure reliable performance and accurate anomaly interpretation.

Cont./nu	0.01			0.05		
Anom. Rate	2%	5%	8%	2%	5%	8%
<i>Baseline:</i>						
OC-SVM	0.565±0.044	0.561±0.026	0.557±0.018	0.667±0.039	0.669±0.020	0.663±0.014
IFO	0.874±0.026	0.882±0.035	0.884±0.036	0.874±0.026	0.882±0.035	0.884±0.036
LOF	0.922±0.013	0.918±0.008	0.918±0.007	0.922±0.013	0.918±0.008	0.918±0.007
<i>Ours:</i>						
OAE	<u>0.992±0.001</u>	<u>0.992±0.001</u>	<u>0.992±0.001</u>	<u>0.992±0.001</u>	<u>0.992±0.001</u>	<u>0.992±0.001</u>
VAE	0.905±0.028	0.899±0.016	0.892±0.027	0.901±0.021	0.899±0.017	0.900±0.014
CAE	0.983±0.004	0.983±0.003	0.982±0.002	0.983±0.005	0.983±0.003	0.982±0.002

Cont./nu	0.10			0.15		
Anom. Rate	2%	5%	8%	2%	5%	8%
<i>Baseline:</i>						
OC-SVM	0.730±0.038	0.734±0.017	0.729±0.012	0.776±0.037	0.782±0.015	0.776±0.012
IFO	0.874±0.026	0.882±0.035	0.884±0.036	0.874±0.026	0.882±0.035	0.884±0.036
LOF	0.922±0.013	0.918±0.008	0.918±0.007	0.922±0.013	0.918±0.008	0.918±0.007
<i>Ours:</i>						
OAE	<u>0.992±0.002</u>	<u>0.992±0.001</u>	<u>0.992±0.001</u>	<u>0.992±0.002</u>	<u>0.992±0.001</u>	<u>0.992±0.001</u>
VAE	0.905±0.021	0.909±0.016	0.900±0.012	0.894±0.028	0.900±0.020	0.907±0.023
CAE	0.982±0.005	0.983±0.003	0.982±0.002	0.982±0.005	0.983±0.003	0.982±0.002

Table 5.12 Presents the AUC performance of baseline models versus the proposed autoencoder-based models, the models were tested under various anomaly rates and contamination parameters using unsupervised learning with 10-fold CV.

Models	p-value	Wilcoxon	Mean diff.	Effect Size (Cohen's d)
<i>Baseline:</i>				
OC-SVM	<0.001	<0.001	0.24	3.22
ISF	<0.001	<0.001	0.04	1.33
LOF	<0.001	<0.001	0.01	1.49
<i>Autoencoders:</i>				
OAE	<0.001	<0.001	0.00	1.04
VAE	<0.001	<0.001	0.05	3.00
CAE	<0.001	<0.001	0.01	2.07

Table 5.13 Summarises the evaluation of semi-supervised and unsupervised models under similar settings (contamination rate of 0.05, anomaly level of 5%, 10-fold CV). The results presented that semi-supervised models achieved higher performance.

5.5 Summary

This thesis addressed the third research objective by developing an anomaly detection model for detecting anomalies in text data. Driven by the research gap in the area of textual anomaly detection, this study proposes several autoencoder architectures integrated with transformer-based embeddings to detect contextual anomalies in trade data transaction. Extensive experimental results presented that the standard autoencoder model, the OAE, was superior to the baseline models in both semi-supervised and unsupervised settings. Markedly, the simplest variant of autoencoders, the OAE, consistently achieves superior performance compared to the baselines and other autoencoder variants in any experimental configurations. Overall, under the unsupervised learning setting, all models exhibited significant performance degradation compared to their semi-supervised counterparts. However, unsupervised learning provides practical advantages, because it is capable to learn trade data that reflects real-world conditions where noise may be present in the training data. The practical contribution of this study lies in its potential real-world application to support authorities in identifying product misclassifications, thus helping the authorities to mitigate associated risks in the trade domain. Additionally, this method empowers improving the reliability of trade data, which can be used for further analytical decision-making.

In the following chapter, we address the challenges associated with bilingual trade

transactions commonly found in international trade data.

Chapter 6

Bilingual Product Classification through Keyword Integration in Low-Resource Trade Data

In this chapter, we turn to the fourth objective of this thesis to demonstrate our approach for classifying bilingual traded product declarations. This study proposes a key novelty approach by developing models capable of handling mixed language in trade transactions, particularly product description elements. The objective is to construct high-performing bilingual models to address the limitations of monolingual approaches, which have dominated existing literature in the trade domain by proposing a method aiming to produce strong bilingual classification performance suitable for low-resource target languages.

6.1 Introduction

Different languages play a significant role in decision-making (Tenzer *et al.*, 2017), especially in understanding regulations and fulfilling compliance obligations in international trade procedures. A study in the international trade domain found that the greater the linguistic barriers between countries, the lower their trade volumes (Lohmann, 2011). Trade declarations, which are standardised documents used for international trade exchange, have fields for inputting product descriptions in free-text format, and it is expected that this text contains product information in languages to comply with the trade authority. However, diverse language backgrounds lead traders to use languages other than the domestic language frequently. They often choose widely spoken languages or industry terms like English, which is common in international trade (Novith, 2024; Paramartha *et al.*, 2021). Fortunately, this practice is permitted under Indonesian domestic regulation to allow traders to use English for

supplementary explanation of product descriptions to ensure accurate trade classification and regulatory compliance (Indonesian Customs, 2009). Similarly, the phenomenon of dual language was found in the Chilean trade dataset, which product descriptions usually contain Spanish terminology as substitutes for equivalent terms of product attributes (de Artiñano *et al.*, 2023). It is essential to acknowledge that bilingual datasets should be taken into account when developing automated product classification systems. These datasets introduce additional requirements necessary to equip models with multilingual features that are beneficial for international trade facilitation (Grainger, 2024).

The occurrence of mixed languages presents unique challenges for automatic classification tasks. This topic is often neglected because most studies evaluate model performance using monolingual datasets, mainly in English, as noted in Chapter 2 (Luppés, 2019; Ruder, 2020; Shubham *et al.*, 2023; Spichakova and Haav, 2020a). This reliance on a single language limits the applicability of such models in multilingual trade environments, where linguistic diversity exists. On the other hand, several studies have utilised non-English datasets, but only to evaluate model performance in specific regional contexts using monolingual datasets, such as those in Brazilian Portuguese, Korean, and Chinese datasets (Lee *et al.*, 2024; Liao *et al.*, 2024; de Lima *et al.*, 2022; Marra de Artiñano *et al.*, 2023). However, it is worth noting that the availability of non-English datasets mentioned in the public domain is limited, primarily due to non-disclosure agreements with data providers.

Studies have investigated the challenges of classifying multilingual inputs and examined multilingual models involving cross-lingual patterns and shared linguistic features (Conneau and Lample, 2019; King, 2024; Lehmann *et al.*, 2020; Pires *et al.*, 2019). Several approaches have been proposed, such as utilising transfer learning to adapt knowledge in a pretrained model to specific downstream tasks, such as multilingual classification tasks (Ahuja *et al.*, 2020; Rao, 2020). Moreover, involving pretrained models based on Transformer architectures has significantly progressed multilingual text classification tasks (Wang and Banko, 2021; Zhang *et al.*, 2022). In the domain of international trade, BERTimbau (Souza *et al.*, 2020), has been fine-tuned using trade data in the Portuguese language for trade product classification (de Lima *et al.*, 2022).

High classification accuracy is important, but it is essential to align text representations of multilingual input as well. Such alignment not only enables downstream tasks but also leads to further improvements in classification performance (Mohammadshahi *et al.*, 2019; Ruder *et al.*, 2019; Schuster *et al.*, 2019). Several approaches have been proposed for aligning multilingual text representations. An implicit alignment technique, exemplified by XLM-RoBERTa, has achieved cross-lingual alignment through training on large-scale multilingual corpora without explicit cross-lingual supervision (Conneau *et al.*, 2020a). This method needs large multilingual datasets, but these are scarce in trade domain. Explicit alignment methods, such as those employed in XLM (Conneau and Lample, 2019), typically involve pairing training samples across languages to facilitate cross-lingual representation learning. This pairwise alignment can be achieved through various strategies. One approach involves pairing the source language data with auxiliary modalities, such as images, to enrich the representation space (Mangrulkar *et al.*, 2022) or, in most cases, cross-lingual approaches rely on pairing source texts with their translated equivalents (Adams *et al.*, 2017; Park *et al.*, 2024). In addition to direct translation, augmentation techniques, such as replacing words with synonyms or semantically related terms, have been found effective in improving classification performance, especially in low-resource and class-imbalanced settings (Bayer *et al.*, 2023; Nair *et al.*, 2024; Xiang *et al.*, 2020). However, the nature of trade descriptions, which often consist of specialised product terminology and include mixed or complex attributes, poses challenges for the application of augmentation techniques. In particular, it is difficult to determine which main product terms should be replaced or retained without compromising the semantic integrity of the description.

The presence of mixed-language product descriptions in trade data (Marra de Artiñano *et al.*, 2023; Paramartha *et al.*, 2021), underlines the necessity to develop models provided with a feature that delivers strong classification performance while enabling effective cross-lingual alignment for multilingual information retrieval. Inspired by cross-lingual representation learning approaches (Conneau and Lample, 2019), the use of parallel text samples across languages emerges as a promising strategy for enhancing multilingual text representation, which supports semantic alignment and improves classification performance across languages. Building on this foundation, we incorporate extracted keywords as auxiliary features to enrich the textual

representation, rather than employing a translation-based technique for data augmentation.

Keyword-based over translation-based approaches

The adoption of a keyword-based method for product classification, rather than relying on machine translation for bilingual classification, was motivated by some considerations. First, translation-based pipelines rely on machine translation systems, which introduce additional system dependencies and computational overhead. The limited capability of such tools in handling the complex and domain-specific characteristics of product descriptions increases the risk of mistranslation. Additionally, trade descriptions frequently contain typographical errors, technical terminology, and industry-specific abbreviations that lack direct equivalents in the target language. In some cases, such expressions are only partially translated, thereby reducing the semantic reliability of the input. This may lead to the loss or distortion of domain-specific terminology and introduce error propagation risks that are subsequently inherited by the classification model. In contrast, a keyword-based method for bilingual classification reduces operational complexity by eliminating reliance on third-party machine translation systems. The proposed approach enables the model to learn language-specific representations while preserving original lexical and contextual information. Cross-linguistic representation is established through bridging the integration of keyword-based features derived from each language. However, the translation-based approach remains a viable baseline for future investigation. In contexts where high-quality trade datasets and robust machine translation systems are available, it becomes feasible to conduct a comparative evaluation between translation-dependent pipelines and keyword-based bilingual models.

The use of pretrained transformer-based language models in this process is motivated by their capacity to encode rich semantic representations, including for the brief and domain-specific nature of trade product declarations and the advantage of utilising pretrained models for text classification. These inputs subsequently function to train neural network-based classifiers to predict product categories under a supervised learning framework.

With the objective mentioned above, it is hypothesised that enriching the original

product description with additional linguistic signals, extracted bilingual keywords, can enhance HS code classification performance in bilingual contexts. Additionally, the impact on model performance is expected to be effective in settings where the target language constitutes a low-resource environment. This approach encourages the model to identify latent product terms that may not be explicitly present in the raw text. By combining the main product description with the product keywords through employing a deep neural network, the system is expected to have the capability to predict product categories in bilingual variations.

Our main contributions to this chapter are as follows:

1. We propose a novel configuration of paired training samples consisting of product descriptions and their corresponding cross-lingual keyword features, referred to as mixed keywords. The keywords were extracted using a simple yet effective method to combine with the main product description. To the best of our knowledge, this is the first study to integrate domain-specific keywords with bilingual trade product declarations to automate HS code classification.
2. We develop a bilingual model architecture that includes several neural network components: a dual-encoder backbone for processing text and keywords separately, a fusion layer for integrating multimodal features, and a deep classification layer for HS code prediction. The proposed model presented improved model performance in bilingual trade classification and enabled cross-lingual representation alignment.
3. We propose a bilingual model trained through incremental training to generate a bilingual model that effectively achieves high performance in low-resource settings. The empirical evaluation compared to several established multilingual baselines shows the effectiveness of our proposed method as a potential application for a real-world system.

The structure of this chapter is organised into four sections. Section 6.2 introduces the proposed methodology for developing a bilingual model, including keyword extraction and keyword classifier, keyword pair configuration, and model architecture. Section 6.3 describes the datasets utilised, experimental settings, and baseline models. Section 6.4 presents experiment results and comparative analysis, reporting model

performance evaluation, embedding alignment assessment, cross-lingual product search case studies, incremental training experiments, and ablation studies. Finally, Section 6.5 summarises the key findings and conclusions of this chapter.

6.2 Methodology

Keywords refer to words or phrases that encapsulate the main topic, theme, or essential information surrounded by a text (Abulaish *et al.*, 2022; Chen, 2024; Nomoto, 2022). The keywords are essential in domain-specific contexts where terminology is often highly specialised (Abulaish *et al.*, 2022; Chen, 2024; Nomoto, 2022). Most terminologies and texts that exist in specialised fields, such as trade, legal, or medical domains, are often complex and ambiguous, making keywords valuable for narrowing the focus to the core concepts (Zhang, Lu, *et al.*, 2021). In classification tasks involving short-texts input formats, the results are often driven by just one or two key terms (Zhou *et al.*, 2023). Therefore, keywords can act as discriminative indicators by directing models toward the most informative features of the text (Blanchard *et al.*, 2022; Chuanakrud *et al.*, 2021).

As such, we propose that integrating keywords with the main text can intensify contextual signals, enrich textual representation, and strengthen model performance. Keywords act as valuable contextual information for trade datasets, since product declarations are typically brief and contain mixed product attributes that need emphasis for effective classification. Integrating keywords into classification tasks should enhance text representation by providing more informative features that improve the model's understanding and performance. However, the effectiveness of the keyword-based method is contingent upon the quality of the keyword extraction process, as imperfect extraction may introduce noisy or inaccurate features. Keywords integration is also expected to preserve important semantic information from minority classes, which could potentially improve model generalisation. However, this approach serves primarily as a feature improvement strategy rather than a primary solution for data imbalance.

In bilingual contexts, using only product class labels for cross-lingual alignment struggles to capture shared semantic relationships because of language-specific differences in trade transaction representation. On the other hand, the use of keywords

becomes promising as they provide higher-level representations, acting as semantic anchors that bridge and generalise beyond product category labels. To address this issue, we propose to extract keywords from the main text and use them as signals to guide the model. The objective is to improve the alignment of product representations across different languages, guaranteeing that products belonging to the same category are similarly represented. This approach differs from previous studies, where keywords were used to replace and represent the original texts (Onan *et al.*, 2016). Our proposed method corresponds with the study of Zhou in strategy to expand input features by integrating both textual data and keywords (Zhou *et al.*, 2023). However, that study employs keywords to retrieve supplementary knowledge from external open knowledge bases (e.g., DBpedia²⁰), which is then fused with the original text to enrich the representation.

Our approach is framed within the scope of feature engineering, where raw textual data is systematically transformed into multiple types of informative representations (Zheng and Casari, 2018). Product descriptions are treated both as primary inputs and as sources of syntactic and semantic information, and keywords are conceptualised as distilled feature representations, encapsulating the core meaning associated with each class label. This approach integrates keywords with the primary product descriptions as input for training bilingual classification models. By leveraging pretrained transformer-based language models, contextual embeddings can be effectively adapted to reflect the characteristics of trade datasets (Arora *et al.*, 2020), equipping the model with the capacity to identify nuanced semantic differences in product descriptions. A neural network-based architecture is employed to learn joint representations from both feature sources, with the objective of producing semantically rich and contextually accurate outputs.

Figure 6.1 illustrates the proposed method, which integrates product descriptions and extracted keywords utilising a deep learning architecture for bilingual trade product classification. The architecture is composed of three core components, where the first two blocks, addressing keyword extraction, keyword modelling and text-keyword samples configuration that discussed in detail in Section 6.2.2. The construction of the

²⁰ <https://www.dbpedia.org/>

bilingual classification model is explained in the following subsections: Section 6.2.3. introduces the dual encoder strategy; Section 6.2.4 discusses the fusion mechanisms for integrating textual and keyword features; and Section 6.2.5 describes the deep classification layer.

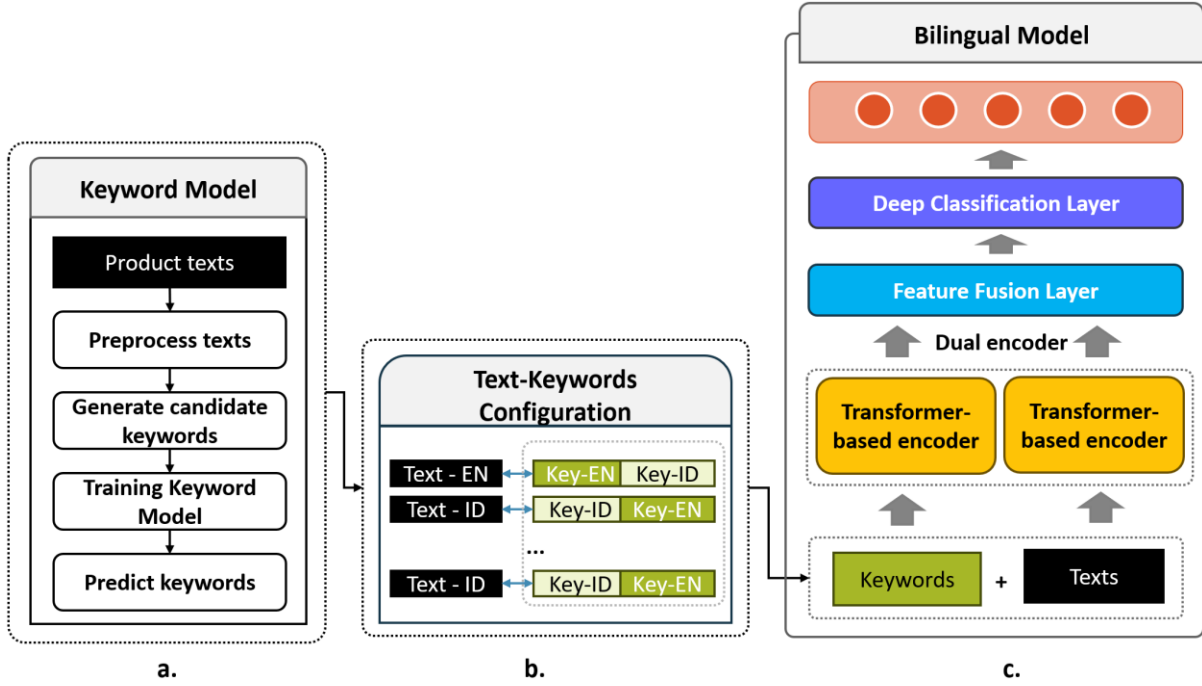


Figure 6.1 Overview of the methodology for text and mixed keyword integration in bilingual product classification, consisting of: a keyword prediction model, sample configuration, and a bilingual classification model for predicting product categories.

6.2.1 Problem Formulation

We formulate the bilingual-text classification problem as follows. Let $L \in \{l_{EN}, l_{ID}\}$ be a set of languages where l_{EN} represents product description in English and l_{ID} represents Indonesian, $X = \{x_1^{g_1}, x_2^{g_2}, \dots, x_z^{g_z}\}$ be a set of product descriptions where $g_i \in L$. Let $C = \{0, 1, \dots, c\}$ be the categories (i.e., HS codes) of product descriptions. The dataset is denoted by $D = \{(x_1^{g_1}, y_1), (x_2^{g_2}, y_2), \dots, (x_n^{g_n}, y_n)\}_{i=1}^n$, which consists of n samples and y_i an associated label where $y_i \in C$. Let $K_i = \{k_1^{g_1}, k_2^{g_2}\}$ Keywords represent each product category, y_i . Given a training dataset $D = \{(X_1, K_1, y_1), (X_2, K_2, y_2), \dots, (X_N, K_N, y_N)\}_{i=1}^N$, the goal of our task is to train a model F that predicts HS code $\hat{y}_i = F((X_i, K_i), \theta)$, where θ , a set of model parameters for a given product with a description X_i and the extracted keywords K_i .

6.2.2 Keyword Extraction and Modelling

Keyword extraction techniques can be divided into two main categories: supervised and unsupervised (Siddiqi and Sharan, 2015). Supervised keyword extraction approaches require keywords to be explicitly labelled in each text sample, which can be implemented using a traditional machine learning or neural network as classifier. Unsupervised keyword extraction works without labelled data, instead of relying on the statistical and structural characteristics inherent in the text sequence. Prior research studies have implemented keyword extraction utilising term-frequency (TF) (Firoozeh *et al.*, 2020), term-importance (Hashemzadeh and Abdolrazzaghi-Nezhad, 2020) or advanced techniques, such as KeyBERT, which is based on Transformer models, capturing contextual semantics for keyword identification (Grootendorst *et al.*, 2023).

Trade product descriptions consist of domain-specific terms designed for international trade contexts, providing detailed product information and specifications. The product description within the same product category (HS code) may vary substantially, let alone if the product description comes from different languages. To address this variability, we begin by extracting keywords that characterise each product category, based on the assumption that frequently occurring terms better represent document content (Firoozeh *et al.*, 2020). Given the nature of trade datasets, we employ TF for keyword extraction, following unsupervised learning. The domain-specific terminologies make frequency-based approaches particularly relevant in trade data because important trade terms naturally occur with higher frequency. The extraction pipeline begins with a pre-processing stage, which includes transforming all characters to lowercase, eliminating duplicate entries, punctuation marks, numerical values, mixed alphanumeric strings, and supplementary product details. Word frequency is then computed for each class, and the highest-ranking term is selected as the representative keyword. Building on the rationale of Zaraini (Zaraini and Yusop, 2025), by utilising limited keywords, assuming that multiple keywords may degrade performance due to increased semantic noise and potential ambiguity. The outcome of this process was a single top keyword for each product category for each language. Finally, each keyword was combined to form two keywords, which we refer to as the Mixed Keywords (MK) strategy. The detailed procedure of the MK configuration is outlined in Algorithm Algorithm 6.1.

MKs were used as labels to annotate the training data used to train a keyword classifier in the subsequent step. This classifier is developed to capture the relationship between product descriptions and their representative keywords, functioning as a contextual anchor to enhance cross-lingual alignment between product descriptions and improve the model's overall classification performance. Following supervised learning to train the keyword classifier, this approach constitutes a discriminative modelling approach where each product description is mapped to a single defined MK label. The keyword classifier involved fine-tuning language models, and in this case, we employed DistilBERT with MK as the labels. In the inference stage, the keyword classifier was utilised to predict MK for product descriptions and pair each product description with its corresponding predicted MK. These pairwise samples were subsequently used to train a bilingual classification model.

Algorithm 6.1: Algorithm for selecting and configuring keyword candidates

```
1: Input:
2:   DocsEN: English description texts of trade transactions
3:   DocsID: Indonesian description texts of trade transactions
4:   product_categories: shared product category labels across languages
5: Output:
6:   Paired of mixed keywords to represent each of the product categories
7: Procedure:
8: Begin
9:   For class in product_categories do:
10:    # Select the keywords in English from each class
11:    VocabEN = {}
12:    For docEN in DocsEN do:
13:      vocabEN ← preprocessing (docEN)
14:      VocabEN ← merge (vocabEN)
15:    End for
16:    KeywordsEN ← sort (VocabEN)
17:
18:    # Select the keywords in Indonesian from each class
19:    VocabID = {}
20:    For docID in DocsID do:
21:      VocabID ← preprocessing (docID)
22:      VocabID ← merge (vocabID)
23:    End for
24:    KeywordsID ← sort (VocabID)
25:
26:    # Sort the most frequent keywords from each class
27:    keywordEN ← sort (KeywordsEN)
28:    keywordID ← sort (KeywordsID)
29:    # Arrange mixed keywords
30:    MixedKeywords[class] ← [keywordEN, keywordID]
31:  End for
32: End
```

6.2.3 Dual Encoder

Following the construction of paired input samples comprising product descriptions and keywords (MK), our next step involved encoding these textual sequences into embeddings. For this purpose, we utilised transformer-based models to generate the contextualised embeddings, specifically utilising DistilBERT due to its balance between computational efficiency and representational performance to handle trade product descriptions. In our configuration, separate encoders were applied to the product description and keyword inputs, where each produced a 768-dimensional embedding vector. The dual-encoder strategy was adopted to allow each encoder to

specialise in its respective input stream, thereby preserving the ability to learn the unique linguistic and semantic characteristics of product descriptions and keywords independently. In the dual-encoder architecture, the product description denoted as X , was encoded into a latent representation, H_T , via the text encoder f_T . Similarly, the keyword sequence, K , was encoded by the keyword encoder, f_K , resulting in the representation H_K . Let n and m denote the maximum sequence lengths of the text and keyword, respectively, and let d represent the embedding dimension. The resulting embeddings are defined as:

$$H_T = f_T(X) \in \mathbb{R}^{n \times d}$$

$$H_K = f_K(K) \in \mathbb{R}^{m \times d}$$

6.2.4 Fusion Layer

For the fusion layer, this thesis adopted an intermediate fusion strategy wherein feature concatenation is performed after the encoding phase, rather than at the raw text level (early fusion) (Boulaiah *et al.*, 2021). This decision is motivated by producing more expressive representations by integrating the distinct latent features of product text and keywords. To accomplish it, the contextual embeddings of the $[CLS]$ token from both the text and keyword encoders are concatenated, resulting in a composite vector h_c . The operation is formalised as follows:

$$h_T = H_T[CLS]$$

$$h_K = H_K[CLS]$$

$$h_c = \text{concat}[h_T; h_K] \in \mathbb{R}^{2d}$$

A FFN was utilised to process the concatenated embeddings from the two encoders (each of 768 dimensions), which projected them into a unified 768-dimensional representation through a series of linear layers. The unified embeddings transformation was processed by layer normalisation, which standardises the feature distributions and contributes to training stability by mitigating issues, such as exploding or vanishing gradients. Rectified Linear Unit (ReLU) was utilised to introduce non-linearity to capture complex interactions between the fused embeddings. The process was then followed by dropout layers, which act as a regularisation mechanism by randomly deactivating neurons during training to reduce the risk of overfitting. The overall process is outlined as follows:

$$\begin{aligned}
f_{fusion}: \mathbb{R}^{2d} &\rightarrow \mathbb{R}^d \\
x &= W_f h_c + b_f \\
x &= \text{LayerNorm}(x) \\
f_{fusion}(h_c) &= \text{ReLU}(x)
\end{aligned}$$

6.2.5 Deep Classification Layer

The unified embeddings were processed utilising a three-layer fully connected classification module, which involved ReLU to model non-linear relationships, dropout regularisation to prevent overfitting and trained using cross-entropy loss to optimise parameters for the classification task.

The model training employed cross-entropy loss with label smoothing by replacing hard target labels with soft probability distributions to reduce overconfident predictions and to improve robustness to noisy labels (Lukasik *et al.*, 2020). The integration of these components in classification layers was assumed to result in a more stable and high-performing model. The classification process is denoted as follows:

$$\begin{aligned}
a^{[0]} &= f_{fusion}(h_c) \\
z^{[l]} &= W^{[l]} a^{[l-1]} + b^{[l]}, \text{ for } l = 1, 2, 3 \\
a^{[l]} &= \text{ReLU}(z^{[l]}) \\
P(y|X, K; \theta) &= \text{Softmax}(z^{[3]})
\end{aligned}$$

6.3 Experiments

This section describes the datasets, experimental settings, baseline models, and evaluation metrics used in this study.

6.3.1 Datasets

Datasets from two different languages were required to build a bilingual model. In this experiment, we used the datasets in English and Indonesian. We utilised publicly available trade transaction datasets in English, which have been referenced in numerous studies (Luppess, 2019; Ruder, 2020; Shubham *et al.*, 2023; Spichakova and Haav, 2020b). These datasets were sourced from publicly available data on the Internet (Enigma, 2018) and (Zaubas, 2018). The Indonesian dataset was sourced from an Indonesian financial institution, which has been cited in the literature (Harsani *et*

al., 2020; Paramartha *et al.*, 2021) and was collected during a field research study approved by the Cardiff University, School of Computer Science and Informatics Research Ethics Committee. A selected product category intersecting with the availability of the Indonesia dataset using six-digit HS codes and the overview of the dataset statistics, including total samples is provided in Table 6.1. This selection results in 49 class categories, with their distribution illustrated along the x-axis in Figure 6.2.

Language	Rows
English	34,937
Indonesian	19,920

Table 6.1 The dataset size in English and Indonesian.

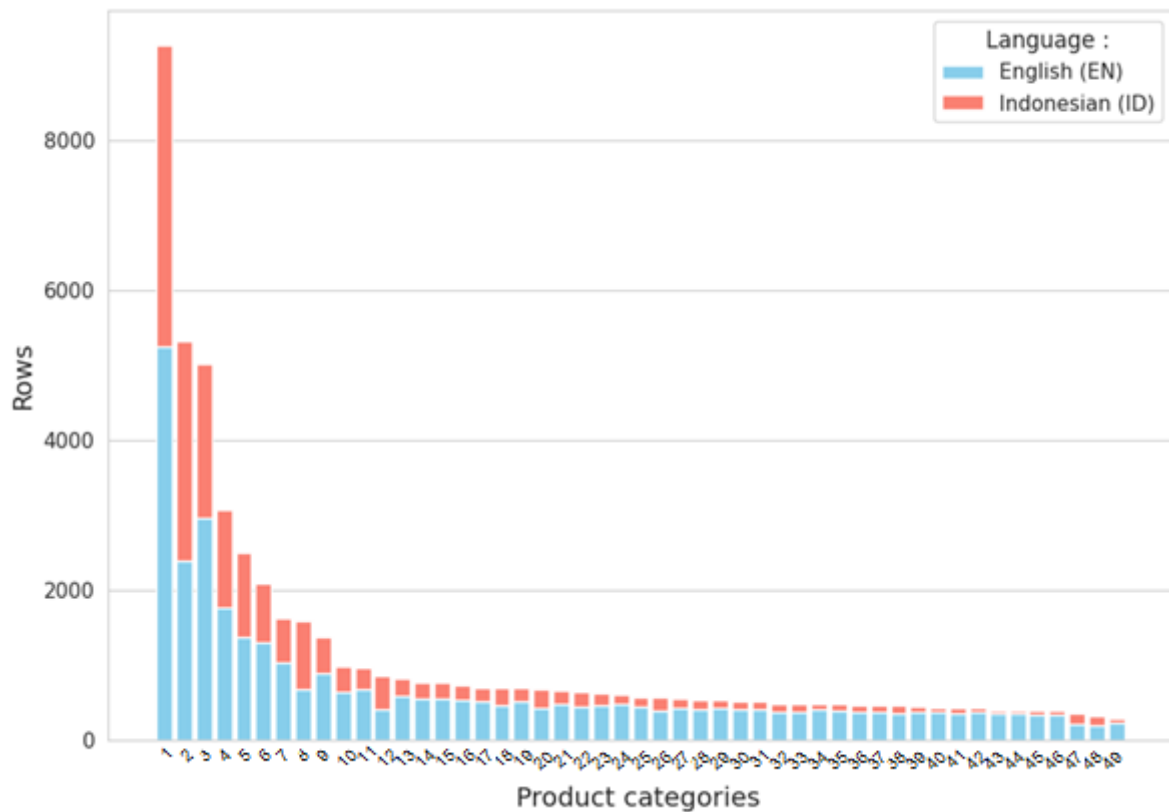


Figure 6.2 Distribution of bilingual data across the 49 HS product categories.

To illustrate the outcomes of our proposed keyword extraction and modelling method, we applied the approach to a subset of the dataset. Table 6.2 presents selected examples of product descriptions in both English and Indonesian, together with their associated HS codes and the corresponding extracted keywords.

No	Text - EN	Text - ID	HS code	Mixed Keywords
1	aj0711 brand reebok os lbt black training apparel women tank top weft knitted 91 pes 9 el	baju kaos rajutan 89 polyester 6polyurethane 5 spandex	610990	['kaos', 'tshirt']
2	lady footwear 317289 brand mochi	1 x the alila sandal 36 vegan	640220	['sandal', 'slipper']
3	Il 149 fb motion 9 insole 1800 pairs	bagian dari sepatu olahraga bottom model36735	640620	['footwear', 'sepatu']
4	gu06 umbrella alluminium with stone base 60 kgs for outdoor furniture	aksesoris payung taman britam 064 38249	660110	['payung', 'umbrella']
5	kwt38 office table with 2 drawers 120cm x 60cm x 76cmisfta conocoisfta165748 dt29062016	sink cabinet berupa lemari dari kayu mahony	940360	['meja', 'table']
6	chair bavne vidle synthetic rattan	meja dan kursi set dari rotan set 3	940381	['rattan', 'rotan']

Table 6.2 Bilingual dataset examples accompanied by their corresponding HS codes and extracted keywords.

6.3.2 Experiments Settings

Model Baselines

This experiment utilised several pretrained multilingual language models as baseline models to evaluate the effectiveness of the proposed method, as follows:

- Multilingual BERT (Devlin *et al.*, 2019): mBERT is an extension of the widely adopted language model, which was pretrained on 104 languages, making it a language-agnostic model that generalises well across multilingual settings.
- XLM-R (Conneau *et al.*, 2020b): XLM-RoBERTa builds upon the RoBERTa architecture, an optimised variant of BERT. This multilingual transformer model was pretrained on text from 100 languages. XLM-R showed superior results on multiple cross-lingual benchmarks in comparison to mBERT.
- Multilingual DeBERTaV3 (He *et al.*, 2023): mDeBERTaV3 is an extension of the DeBERTa architecture designed to improve multilingual understanding. It

achieved significant performance gains, surpassing XLM-R by 3.6% across 15 languages.

- Sentence BERT (Reimers and Gurevych, 2019): SBERT generates sentence-level embeddings by leveraging BERT in its architectural network. For this experiment, one of SBERT's off-the-shelf models, paraphrase-multilingual-MiniLM-L12-v2, was employed as a baseline model to represent multilingual sentence embeddings.
- IndoBERT (Koto *et al.*, 2020): IndoBERT is a transformer-based language model that was pretrained exclusively on Indonesian corpora, making it a suitable baseline for text classification tasks involving the Indonesian language. IndoBERT is designed to capture linguistic features of Indonesian language.

Experimental Setup

All experiments and data analysis procedures were executed on P100 GPUs with 16 GB of RAM on computing facilities at Cardiff University, managed by Advanced Research Computing at Cardiff (ARCCA). All pretrained models are obtained via the HuggingFace²¹ library (Wolf *et al.*, 2020), leveraging its extensive repository of state-of-the-art transformer-based models and the models are preferred over the base version and uncased models. The embedding size of all the models is set to 768, with the input text or keyword limited to a fixed sequence length of 128 tokens per sample initiating truncation and padding process to reach this length and the batch size for training process employs a learning rate of 2×10^{-5} and is conducted over three epochs as sufficient numbers for fine-tuning in most tasks with optimisation employs the cross-entropy loss function in conjunction with the Adam optimiser (Devlin *et al.*, 2019). This experiment adopts a dropout rate of 0.2, consistent with previous research (Li, Ma, *et al.*, 2022), as it provided a reliable performance across experiments. The summary of our configuration for the architecture is seen in Table 5.5.

²¹ <https://huggingface.co/>

Hyperparameter	Value	Hyperparameter	Value
Embedding dimension	768	Learning rate	2×10^{-5}
Max length tokens	128	Dropout	0.2
Epoch	3	Optimizer	Adam
Batch size	16	Activation Function	ReLU
Classification head	768-512-256	Label smoothing	0.1

Table 6.3 Hyperparameter configurations used in the experiments.

Evaluation Metrics

This thesis employs a set of widely adopted metrics: accuracy, weighted precision, weighted recall, and weighted F1-score to evaluate model performance comprehensively. The evaluation strategy in this thesis was conducted to compare with baseline models using a 10-fold CV for model performance evaluation. A simulation on a low-resource setting was performed by varying the availability of target-language data, specifically limiting Indonesian training samples into four levels of samples from 0, 1,000, 5,000, and 10,000. The experimental design reflects a realistic scenario where non-English data is scarce in bilingual model training and evaluating model performance under these configurations by utilising a train-test-holdout split strategy.

In addition to classification metrics, the model assessment also evaluates the quality of cross-lingual embedding alignment. For this purpose, cosine similarity is used to assess the degree of semantic alignment between generated embeddings of trade product descriptions in different languages that correspond to the same HS code. Cosine similarity scores range from 0 to 1, where a score of 1 denotes perfect alignment, indicating that the model has effectively learned cross-lingual representations. The similarity between two embedding vectors x and y is computed using the normalised cosine similarity metric as follows:

$$sim_{cos}(x, y) = \frac{x^T y}{|x|_2 |y|_2} \in [0, 1]$$

An ablation study was performed to evaluate individual component contributions utilised in the proposed bilingual model. The model's performance was assessed by

training separately on samples consisting of product descriptions and keyword representations, respectively. This experimental design aimed to provide empirical insights into how each text component contributes to prediction accuracy and representation quality in bilingual product classification.

6.4 Results and Discussion

6.4.1 Model Performance

The proposed method was evaluated in comparison to baseline models for the classification task, as presented in Table 6.4. The experiments were performed using 10-CV to validate robustness and reliability. The proposed model achieved strong model performance of 97.0% on accuracy and an F1-score of 96.8%. The proposed method presented a significant performance gain, achieving an average improvement of 17.50% in accuracy and significantly outperformed all baselines. In comparison to the best baseline model performance, mBERT, the improvement in F1-score was statistically significant with a large effect size, as seen in Table 6.5.

The effectiveness of the proposed method underlines the advantages of utilising keyword-based features for bilingual classification tasks, where the research findings align with previous studies (Onan *et al.*, 2016; Zhou *et al.*, 2023). However, our proposed method advances the contribution by integrating both the main text and the associated keywords, achieving a significant improvement in bilingual classification performance.

Models	Accuracy	Precision	Recall	F1-Score
<i>Baseline:</i>				
Multilingual BERT	0.820 ± 0.006	0.819 ± 0.007	0.820 ± 0.006	0.817 ± 0.007
XLM-R	0.802 ± 0.007	0.801 ± 0.008	0.802 ± 0.007	0.798 ± 0.007
Multilingual DeBERTa	0.794 ± 0.008	0.793 ± 0.008	0.794 ± 0.008	0.788 ± 0.008
SBERT	0.794 ± 0.004	0.792 ± 0.005	0.794 ± 0.004	0.788 ± 0.005
Indo BERT	0.791 ± 0.005	0.790 ± 0.005	0.791 ± 0.005	0.787 ± 0.006
<i>Proposed method:</i>				
MK + DistilBERT	0.970 ± 0.003	0.969 ± 0.003	0.970 ± 0.003	<u>0.968 ± 0.003</u>

Table 6.4 Complete 10-fold cross-validation (CV-10) test results of all models. The proposed models consistently outperform baseline models across evaluation metrics.

Proposed Model	Baseline Models	p-value	Mean Difference	Effect Size (Cohen's d)
MK + DistilBERT	Multilingual BERT	< 0.001	0.15	24.90
	XLM-R	< 0.001	0.17	25.40
	Multilingual DeBERTa	< 0.001	0.18	24.18
	SBERT	< 0.001	0.18	42.34
	Indo BERT	< 0.001	0.18	32.65

Table 6.5 Statistical comparison between the proposed MK + DistilBERT model and various multilingual baseline models.

6.4.2 Embeddings Alignment

This subsection visualises model embeddings via PCA to explore cross-lingual alignment, considering the limitations of this technique in capturing reduced high-dimensional embeddings. As shown in Figure 6.3, the left panel of the figure displays the embeddings produced by fine-tuned XLM-R, one of the baseline models, which taken from the final hidden layer representing product descriptions across two languages. The data points appear scattered, with representations from both languages intermixed and exhibiting no well-defined clusters. On the other hand, the embeddings generated with the proposed method present a high degree of overlap between data points across languages to indicate potential cross-lingual alignment, as illustrated in the right panel of Figure 6.3.

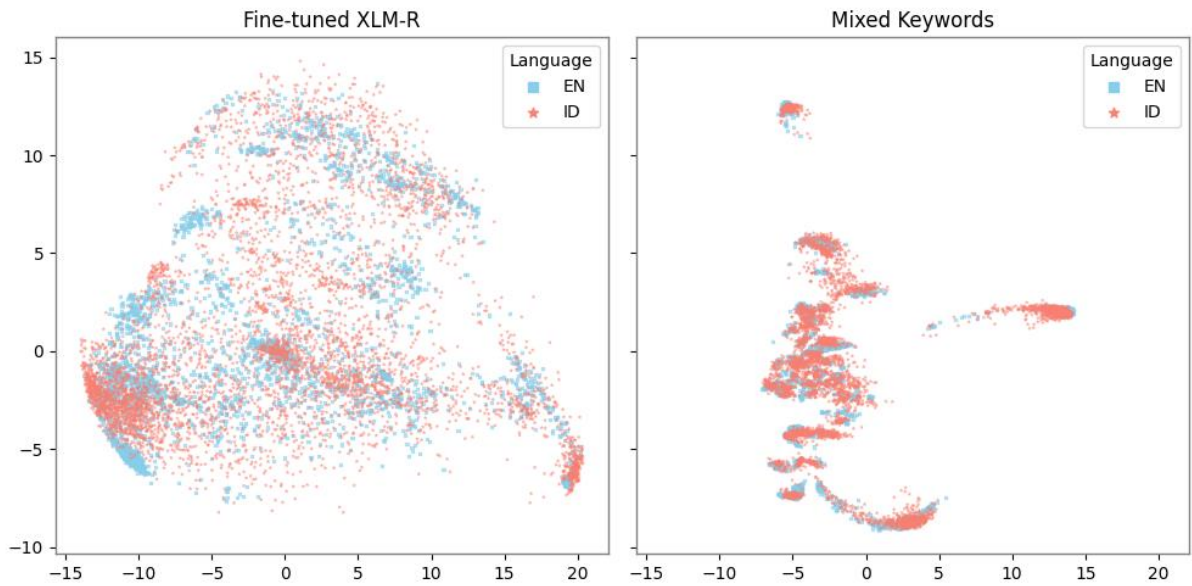


Figure 6.3 Embedding space visualisation of Indonesian (ID) and English (EN) trade data using PCA. The left panel represents embeddings produced by the fine-tuned multilingual XLM-R model and the right panel shows our proposed MK method.

A quantitative analysis of cross-lingual alignment computed the similarity between embeddings in each language across all class categories, complemented with a visualisation. A heatmap was utilised to depict the average embedding similarity score for each class category, as presented in Figure 6.4. This visualisation captured the degree of alignment between embeddings from different languages representing the same HS code. The average similarity between cross-lingual embeddings was computed using cosine similarity, as shown in Evaluation Metrics in section 6.3.2.

The diagonal elements of all matrices denote the same class category of two languages, the darker the colour (dark blue). A comparison between the fine-tuned multilingual pretrained XLM-R and the MK approach was conducted, as illustrated in Figure 6.4. The MK approach achieved the averaged intra-class similarity of 0.92, indicating a high cross-lingual alignment compared to XLM-R that achieved similarity score of 0.64. This enhanced alignment provides a more meaningful representation, revealing that the incorporation of the keyword method encourages product descriptions of the same category to be closely mapped in the embedding space, resulting in higher similarity scores.



Figure 6.4 Heatmap showing similarity scores between inter-class and intra-class labels across Indonesian (y-axis) and English (x-axis) embeddings, generated by the fine-tuned XLM-R and the proposed method. The diagonal represents identical class labels across both languages. Darker colours indicate higher similarity, with dark blue regions highlighting strong alignment between the same class labels in the two languages.

To evaluate the effectiveness of the proposed method used for bilingual product search, a case study was conducted using selected query examples. The queries included the English term "*rattan table*" and the Indonesian term "*Meja Kayu*" (translated as "*wooden table*"). By computing embedding similarity distances, the model identified the two closest products for "*Rattan Table*" in both languages, matched to HS code 940381 (World Customs Organization, 2012), representing Furniture of other materials, including bamboo and rattan, as seen in Figure 6.5 a. Similarly, a query "*Meja Kayu*" returned relevant products under HS code 940360, which corresponds to wooden furniture of a kind used in the bedroom, as illustrated in Figure 6.5 b.

```

=====
Query: rattan table
=====

Top 2 most similar sentences in EN corpus:
940381 : rattan bar set model no8361 1 set with 3 pcs (Score: 0.8470)
940381 : day bednyaman synthetic rattan (Score: 0.8469)

Top 2 most similar sentences in ID corpus:
940381 : tempat lampu rottan kombinasi besi &kain (Score: 0.8437)
940381 : top meja rotan rangka besi (Score: 0.8431)
a.

=====
Query: meja kayu
=====

Top 2 most similar sentences in EN corpus:
940350 : sofa 1pcs made of woodfabricfoam (Score: 0.7565)
940350 : donnut king size bed 198 x 183 2f made of solid wood (Score: 0.7534)

Top 2 most similar sentences in ID corpus:
940350 : newport dresser / lemari pakaian dari kayu mahoni (Score: 0.7746)
940350 : mahogany wooden bed, headboard, dresser/ranjang, kepala ranjang, lemari
pakain dari kayu mahoni (Score: 0.7709)
b.

```

Figure 6.5 Retrieved product descriptions using bilingual queries. Despite variations in the English and Indonesian input terms, the model consistently retrieves products with matching HS codes, demonstrating its effectiveness in processing bilingual queries and aligning semantically equivalent descriptions.

The bilingual product search results present that the proposed method provided accurate cross-lingual product recommendations and effectively captured material-specific information embedded in the queries, even when the inputs are minimal. The proposed model has the capability to process both English and Indonesian queries to retrieve the product category, underlining the improvement of the model's understanding of cross-lingual product representation.

6.4.3 Ablation Study

To evaluate the contribution of integrating text and keywords in our model architecture,

we conducted a comparative analysis between the proposed method, MK, and two model variants: the model trained only with product descriptions (Product Only, PO) and the model trained only with keywords (Keyword Only, KO) for bilingual product classification. The results show that the combined features of the MK model achieved model improvements in all metrics, ranging from 15.87% to 16.04% on average across various model configurations, as shown in Table 6.6. The model performance significantly outperformed the PO model, underlining the effectiveness of keyword integration. In comparison to the KO model, which relies solely on keywords to represent the transaction, the MK model still achieves superior performance, with all metrics improving, ranging from 6.84% to 13.62%. These results provided convincing evidence that utilising keywords independently is insufficient to obtain high classification performance, underlining the benefits of combining them with the main text.

	Accuracy	Precision	Recall	F1-Score
<i>Models:</i>				
Products Only (PO)	0.816 ± 0.005	0.815 ± 0.005	0.816 ± 0.005	0.813 ± 0.005
Keywords Only (KO)	0.904 ± 0.000	0.837 ± 0.001	0.904 ± 0.000	0.865 ± 0.001
Mixed Keywords (MK)	0.970 ± 0.003	0.969 ± 0.003	0.970 ± 0.003	<u>0.968 ± 0.003</u>
<i>Performance Gains:</i>				
MK vs. PO	+15.88%	+15.87%	+15.88%	+16.04%*
MK vs. KO	+6.83%	+13.62%	+6.83%	+10.65%*

Table 6.6 Performance results from the ablation study, comparing the Products Only (PO) and Keywords Only (KO) models. The bottom section reports the performance gains of the proposed Mixed Keyword (MK) method over each variant.

6.4.4 Model Performance in Low-Resource Setting

Access to non-English trade datasets in the public domain is limited, which is considered low-resource data. Within the scope of this research, low-resource data does not imply the complete absence of data. Rather, it refers to a setting in which a limited sample size hinders effective model training. In response to this issue, this study conducted a low-resource experiment in which the bilingual model was trained using a stepwise sampling strategy to assess its classification performance under

constrained data conditions. The proposed model was trained independently on English trade data and mixed keywords, achieving an F1-score of 92.82% in the absence of Indonesian training data, as illustrated in Figure 6.6. Introducing a stepwise addition of Indonesian trade data samples, ranging from 1,000 to 10,000, demonstrates a considerable improvement in classification performance, achieving an F1-score of 96.22%. This result underscores the efficacy of the proposed method in addressing low-resource data and adapting to additional data training to enhance predictive accuracy.

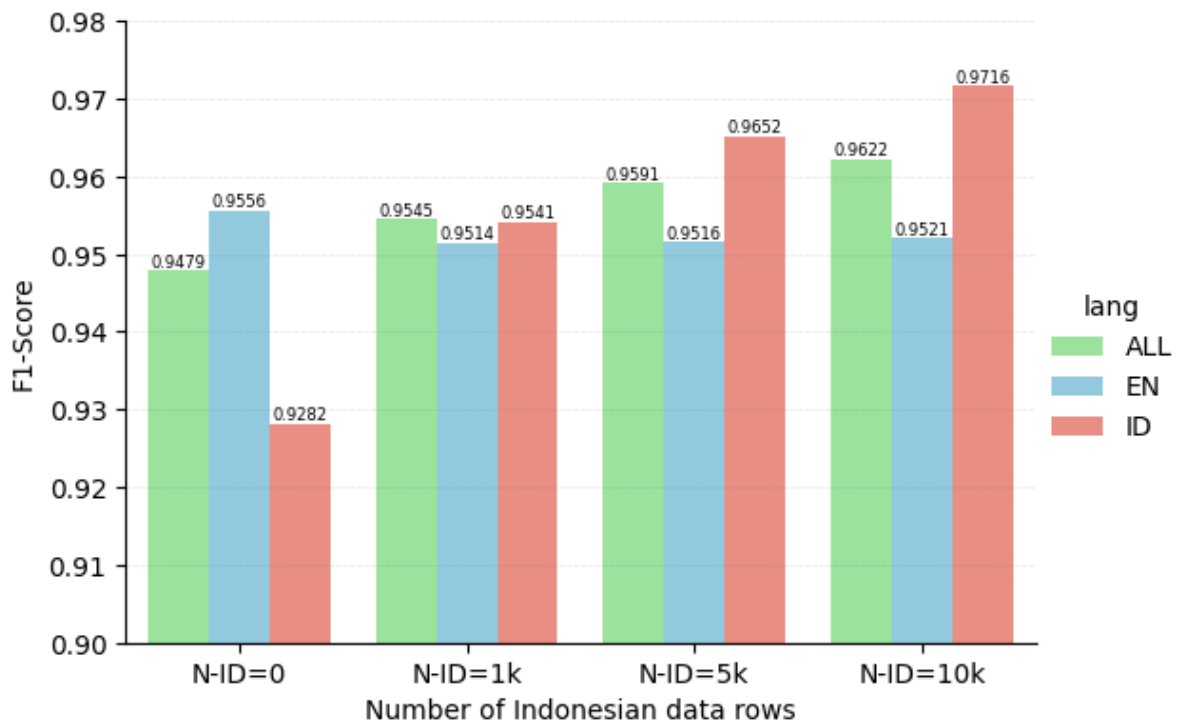


Figure 6.6 Performance of the bilingual model trained with a gradually increasing number of Indonesian training samples. ALL indicates the model's performance on the full bilingual test set, EN represents performance on English test data, and ID reflects performance on Indonesian test data.

This thesis reveals that DistilBERT, which is considered a monolingual language model, can surprisingly perform bilingual product classification. This finding is particularly valuable for low-resource language scenarios, where DistilBERT's greater computational efficiency over larger multilingual models provides practical advantages. Our results are consistent with prior work showing that DistilBERT can effectively handle low-resource language tasks (Cruz and Cheng, 2020) with fine-

tuning approaches applied to optimise model performance (Li, Li, *et al.*, 2020). The cross-lingual transfer methods using fine-tuned multilingual pretrained language models show limited effectiveness when applied to domain-specific trade datasets across different languages. On the contrary, the empirical result suggests that integrating the primary product texts with bilingual MK offers an effective cross-lingual strategy for aligning representations between high-resource and low-resource languages. The strong performance of our proposed method on the trade dataset can be attributed to the role of keywords as language-agnostic semantic anchors. These anchors enable more effective cross-lingual knowledge transfer by leveraging models trained on well-resourced but out-of-domain languages, thus maximising information extraction from keywords even when training data is limited. The results further support the application of keyword integration in text classification tasks, indicating that the mixed-keyword strategy offers promising few-shot learning capabilities and plays a critical role in improving classification performance.

6.5 Summary

To conclude, this thesis addresses the challenge outlined in the research objective by proposing a bilingual classification approach that incorporates keyword features into the model architecture. The keywords integration in the bilingual models enables the capture of domain-specific terminology representation across languages, making the model more robust to bilingual language trade variations. By extracting the mixed keyword and combining it with the trade product declaration, the experimental results demonstrate notable improvements in bilingual classification performance compared to well-established baseline methods. The integration of deep learning components enhances the alignment of product representations across languages, resulting in more representative embeddings for semantically similar product categories, as evidenced by the cross-lingual product search case study. Importantly, the model also performs well in low-resource scenarios, effectively classifying bilingual product descriptions with limited target-language data. Given these findings, the proposed approach holds strong potential for real-world applications, offering both methodological and practical contributions. A practical bilingual classification system can be developed using our approach to assist traders, particularly in Indonesia. This application would offer integrated HS code prediction and cross-language product

search functionality for both English and Indonesian.

Chapter 7

Conclusion

This chapter serves three main purposes. First, it summarises the key findings of the thesis, highlighting that the thesis has addressed the research objectives, as recapitulated in Section 7.1. Second, it outlines the main contributions of the thesis within the academic and practical contexts, in Section 7.2. Finally, Section 7.3 outlines directions for future research, building on the findings presented to advance automated trade document processing capabilities.

7.1 Revisiting Automation Tasks in International Trade

This thesis examines how the potential of deep learning and modern language models has been applied to address the challenges of important trade processes in international trade. By developing and evaluating a range of deep learning approaches, this study has sought to contribute to the fields of machine learning and NLP, as well as the practice of international trade processes.

Chapter 3 highlights the persistent challenge posed by the inherent complexity of the HS classification system, particularly due to the large number of product categories and the terminological gaps between industry usage and official HS guidance. As stated in RQ1, this chapter investigates the effectiveness of a classification pipeline incorporating transformer-based language models and deep neural networks in classifying international trade products into HS codes within the given constraints. Our empirical results show that, when applied to trade product descriptions containing ungrammatical expressions and mixed attributes, transformer-based models are more sensitive to preprocessing than traditional language models and non-contextualised embeddings. The findings indicate that excessive preprocessing tends to negatively impact the model classification performance. Keeping product attributes that appear insignificant can improve model accuracy. Despite of this sensitivity, transformer-

based models consistently outperform traditional language models and those based on non-contextualised embeddings. The key strength is handling out-of-vocabulary terms and typographical errors, which are common in trade data. BERT and DistilBERT yield the most effective results among state-of-the-art transformer-based architectures tested for classification tasks with DistilBERT excelling in model size, faster convergence, and inference. These findings suggest that only specific pretrained language model architectures are well-suited to HS classification tasks, where the performance gap between the highest- and lowest-performing transformer-based models was 7%. Furthermore, in classification problems involving hierarchical taxonomies such as the HS, hierarchical classification approaches yield significantly better performance than flat classification methods, which are more commonly applied in existing trade product classification research.

Chapter 4 addresses RQ2, which examines how representation learning techniques can be leveraged to address challenges such as semantic overlap and anisotropy in trade product descriptions and enhance classification performance. The chapter presents that enhancing the quality of sentence embeddings significantly improves the accuracy of trade product classification. The results indicate that sentence-level embeddings offer a rapid and flexible foundation for machine learning tasks, but off-the-shelf embeddings from models such as SBERT, which are commonly applied in studies of trade-related tasks, are prone to high anisotropy. A similar limitation is found in embeddings derived directly from the [CLS] token of transformer-based models when used without fine-tuning. The findings present that fine-tuning sentence embeddings with supervised contrastive learning through a customised Siamese network effectively reduced anisotropy and improved the representational structure of trade products in the embedding space. This approach contributes to more uniformity of representations. The intra-class similarity scores increased, and inter-class similarity decreased, resulting in higher-quality and semantically consistent embeddings. These enhanced sentence embeddings manifested in stronger product classification performance. Traditional machine learning algorithms used in the experiments, SVM and RF, outperformed the fine-tuned transformer models. RF achieved superior accuracy, though required longer training time than SVM when utilising the generated sentence embeddings. Further analysis also reveals that embedding quality improves with a larger number of positive training samples and

longer training epochs.

Chapter 5 focuses on RQ3, investigating how text-based anomaly detection techniques can be applied within neural network architectures and language models to detect potential product misclassifications. The chapter underlines the findings, showing that a hybrid approach combining contextualised embeddings with autoencoders is highly effective for text-based anomaly detection. Empirical findings confirmed that the standard autoencoder (OAE) outperforms both VAE and CAE under a semi-supervised learning paradigm. The proposed model achieved lower reconstruction errors and surpassed well-established anomaly detection algorithms, including OC-SVM, ISF, and LOF. The results further suggested that when contextualised embeddings from transformer-based models were used, modifications in hidden layer configurations had minimal impact on performance. By contrast, performance declined when traditional or non-contextualised embeddings were utilised. Additionally, the OAE presents the model robustness in unsupervised settings, maintaining reliable detection despite noisy data being present in the training split. These findings suggest that contextualised embeddings combined with OAE provide a potential solution for detecting misclassification in trade transactions.

Chapter 6 presents the performance of the proposed bilingual model in classifying trade data across two languages. The chapter addresses RQ4, which investigates how keyword-based features can strengthen semantic alignment and improve classification performance in bilingual, low-resource international trade settings. It was evident that using a method that combines keywords extracted from each product category in both languages with the main text embeddings resulted in superior classification results. The results presented that the mixed-keyword approach encourages embedding alignment across languages, enabling product descriptions expressed in different languages in the same HS code to be mapped closely in the embedding space. Moreover, the result produces the same product codes when queries are made in different languages. An ablation study was conducted to validate the proposed method. It was evident that combining keywords with the main text consistently outperforms the model performance based only on the main text or keywords in our experiment. Additionally, our proposed bilingual model presented strong classification performance in low-resource settings, where the target language

was limited in training samples.

However, this thesis recognises certain limitations, as the proposed methods are designed with flexibility but may require adaptation to specific jurisdictions and data environments. First, jurisdiction-specific HS extensions limit direct cross-country applicability. While the HS is standardised at the six-digit level, certain customs administrations implement national extensions beyond this level to address domestic regulatory requirements. Consequently, deployment within these jurisdictions would necessitate adaptation through fine-tuning using locally defined HS classifications. Second, variations in trade data quality and descriptive habits may influence model performance and transferability. Differences in language usage, technical terminology, abbreviation usage, and noise levels across datasets can affect semantic consistency and representation learning. Therefore, both the availability and quality of trade data play an important role in determining the effectiveness and generalisability of the proposed models.

7.2 Contributions

This thesis contributes to the field of machine learning for international trade by developing methodologies for product classification, detection of product misclassification, and bilingual product classification. Several aspects of this research have been published in academic papers. The main contributions of this thesis include:

1. Improving predictive accuracy for the trade product classification task through the development of a comprehensive text classification pipeline. This product classification pipeline addresses the distinctive challenges of processing product descriptions by empirically evaluating text preprocessing strategies while accounting for the heterogeneous attributes frequently observed in trade texts. The proposed method has successfully identified the language model architectures most suitable for size, training efficiency, and predictive accuracy by fine-tuning transformer-based pretrained language models using trade datasets. Subsequently, the proposed method integrates the embeddings into a novel hierarchical classification framework, extending the work beyond the flat classification approaches predominantly used in prior studies. The proposed method utilised an explainability tool to highlight text segments that are most

relevant to product predictions, enhancing model interpretability. Collectively, the empirical contribution of the proposed method achieves up to a 22% accuracy improvement over existing methods on public datasets containing more than 3,000 product categories. The methodological novelties presented in this thesis make the proposed methods viable candidates for practical implementation in real-world trade classification applications, supporting traders in classifying products accurately while avoiding penalties and other adverse consequences.

2. This thesis contributes to the development of models to generate enhanced sentence-level representations used for trade product classification. The approach enhances pretrained language models by implementing a Siamese network trained using supervised contrastive learning methodology. Several key components contribute to the result achievement, including paired training data configuration, a task-specific loss function, and selected components used for the model architecture. The contribution of the proposed method reduces anisotropy in the embedding space, enhances semantic coherence by increasing similarity among descriptions within the same product category, and decreases similarity across different categories. The enhanced representation learning strengthens product classification performance. The proposed algorithm has been validated on two publicly available real-world trade datasets, presenting its empirical contribution. The practical contribution of this thesis lies in providing sentence-level embeddings that not only reduce the time required for developing classification models but also improve the accuracy of HS code predictions, supporting traders in accurately classifying products even when dealing with subtle differences between product categories.
3. This thesis contributes to the improvement of product misclassification detection in trade transactions by developing a text anomaly detection method based on autoencoder architectures. The approach introduces three specialised configurations, standard autoencoders (OAE), VAE, and CAE, to capture latent representations of trade text data. A further contribution lies in the integration of contextualised embeddings from transformer-based models with each autoencoder, resulting in a hybrid method capable of identifying contextualised anomalies. The proposed method was designed to operate effectively under semi-

supervised and unsupervised settings, where only normal data are available for training in semi-supervised settings, and training data include both normal and atypical samples for unsupervised learning. Our method shows strong results in detecting anomalies, performing better than the well-established anomaly detection models. In practice, this approach potentially supports trade authorities in spotting product misclassifications and reducing compliance risks. Additionally, the proposed method indirectly contributes to improving data quality, which strengthens the reliability of trade statistics and supports more effective revenue collection.

4. This thesis introduces a novel methodological approach to bilingual product classification by incorporating a mixed keyword pairing mechanism. An approach that integrates keywords as part of data augmentation and feature engineering manifests by proposing an algorithm used for keyword extraction and keyword prediction. The configuration of samples consists of both the main product description and the keywords used to train the bilingual model, where this approach promotes an effective cross-lingual embedding alignment across languages. This design offers a simple yet effective means of enhancing bilingual model performance when compared with the implementation of fine-tuned multilingual models on bilingual datasets. Furthermore, the proposed method leverages the integration of source-target language keywords to address the challenges of low-resource target languages. A practical implementation of this bilingual classification system has the potential to support traders, particularly in the context of Indonesian international trade, which represents the scope of our study. Nevertheless, the method is straightforward to implement in other trade authorities. The system would enable more reliable HS code prediction in bilingual contexts, along with product search functionality across languages.

7.3 Future Research

Future research could further expand this thesis by exploring the capabilities of large language models (LLMs), which have already been discussed in the literature review, regarding Transformer decoder modules for predicting product categories, utilising GPT-3 and GPT-3.5 with a prompt-based approach (de Artiñano *et al.*, 2023; Hussain

and Ahmed, 2023; Navasardyan, 2024). However, several concerns had been identified in the studies, including inconsistent outputs, data privacy issues, and high implementation costs. Nevertheless, recent advances in open-source and offline LLMs offer promising alternatives, although hallucination risks remain a challenge (Huang *et al.*, 2025). Some language models like LLaMA (Grattafiori *et al.*, 2024; Touvron *et al.*, 2023) and DeepSeek (DeepSeek-AI *et al.*, 2025) present advantages in performing few-shot and zero-shot classification. These models also offer explainability and reasoning, and scalability without recurring API fees. Furthermore, these models allow for fine-tuning for domain adaptation with some options, including a full fine-tuning method for maximum adaptation, parameter-efficient approaches (Hu *et al.*, 2021), and few-shot prompting with minimal resources (Wei *et al.*, 2023). However, challenges remain, as adopting open-source and offline LLMs for international trade applications is typically hindered by the requirement for large, curated datasets and high computational power. Resource constraints in some trade authorities further limit the choice between developing in-house models or adopting commercial solutions, which require consideration related to model performance, cost, and data privacy.

The deployment of the systems proposed in this thesis in the infrastructure of trade authorities highlights substantial distinctions between lightweight pretrained language models, such as DistilBERT, and large-scale language models. The proposed systems developed in this thesis represent a practical solution for near-term adoption, owing to their modest computational requirements, which permit deployment on existing data centres or mid-range GPU servers. In contrast, locally deployed large-scale LLMs require more demanding infrastructure, including high-performance multi-GPU clusters and considerable system memory. Such requirements are likely to exceed the current capacity of trade facility environments, for instance, within the infrastructure of the Indonesian trade authority. Consequently, lightweight architectures appear more immediately compatible with existing trade authority infrastructure. The implementation of the proposed information systems is expected to optimise the timeliness and effectiveness of trade processes.

Additional avenues for future research arise from the findings and limitations identified in this thesis. First, future studies could investigate translation-based baselines for bilingual HS code classification. While this thesis adopts a keyword-based integration

approach, an alternative strategy would involve translation-based pipelines to evaluate the trade-offs between computational cost and classification performance, particularly in low-resource language settings. Second, continual learning offers a promising approach to address concept drift in international trade data. As product descriptions and terminology evolve, static models may degrade in performance. Incremental learning frameworks that enable updates without full retraining could enhance robustness and support long-term deployment in operational customs environments. A final avenue for future research concerns model explainability. Since the incorporation of human-in-the-loop mechanisms remains essential, the integration of explainability techniques could strengthen user trust, support informed decision-making, and improve the handling of ambiguous or anomalous cases to ensure regulatory compliance and mitigates potential trade-related risks.

Bibliography

- A. Kumar Singh, Sahu, R. and Ujjwal, K. (2003), "Decision Support System in Customs Assessment to Detect Valuation Frauds", *IEMC '03 Proceedings. Managing Technologically Driven Organizations: The Human Side of Innovation and Change*, presented at the *IEMC '03 Proceedings. Managing Technologically Driven Organizations: The Human Side of Innovation and Change*, pp. 546–550, doi: 10.1109/IEMC.2003.1252333.
- Aamir, M., Mohd, N., Fazli, N. and Hairulnizam, W. (2021), "A Deep Contractive Autoencoder for Solving Multiclass Classification Problems", *Evolutionary Intelligence*, Springer Berlin Heidelberg, Vol. 14 No. 4, pp. 1619–1633, doi: 10.1007/s12065-020-00424-6.
- Abdolshah, M., Teimouri, M. and Rahmani, R. (2017), "Classification of X-Ray images of shipping containers", *Expert Systems with Applications*, Vol. 77, pp. 57–65, doi: 10.1016/j.eswa.2017.01.030.
- Abulaish, M., Fazil, M. and Zaki, M.J. (2022), "Domain-Specific Keyword Extraction Using Joint Modeling of Local and Global Contextual Semantics", *ACM Trans. Knowl. Discov. Data*, Vol. 16 No. 4, p. 70:1-70:30, doi: 10.1145/3494560.
- Adams, O., Makarucha, A., Neubig, G., Bird, S. and Cohn, T. (2017), "Cross-Lingual Word Embeddings for Low-Resource Language Modeling", in Lapata, M., Blunsom, P. and Koller, A. (Eds.), *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, presented at the *EACL 2017, Association for Computational Linguistics, Valencia, Spain*, pp. 937–947.
- Aggarwal, C.C. and Zhai, C. (2012), "A Survey of Text Classification Algorithms", in Aggarwal, C.C. and Zhai, C. (Eds.), *Mining Text Data*, Springer US, Boston, MA, pp. 163–222, doi: 10.1007/978-1-4614-3223-4_6.
- Agrawal, S. and Agrawal, J. (2015), "Survey on Anomaly Detection using Data Mining Techniques", *Procedia Computer Science*, Vol. 60, pp. 708–713, doi: 10.1016/j.procs.2015.08.220.

- Ahuja, A., Rao, N., Katariya, S., Subbian, K. and Reddy, C. (2020), "Language-agnostic representation learning for product search on e-commerce platforms".
- Ait-Saada, M. and Nadif, M. (2023), "Unsupervised Anomaly Detection in Multi-Topic Short-Text Corpora", in Vlachos, A. and Augenstein, I. (Eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, presented at the EACL 2023, Association for Computational Linguistics, Dubrovnik, Croatia, pp. 1392–1403, doi: 10.18653/v1/2023.eacl-main.101.
- Altaheri, F. and Shaalan, K. (2020), "Exploring Machine Learning Models to Predict Harmonized System Code", in Themistocleous, M. and Papadaki, M. (Eds.), *Information Systems*, Springer International Publishing, Cham, pp. 291–303, doi: 10.1007/978-3-030-44322-1_22.
- Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., et al. (2021), "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions", *Journal of Big Data*, Springer Science and Business Media LLC, Vol. 8 No. 1, doi: 10.1186/s40537-021-00444-8.
- Amel, O., Stassin, S., Mahmoudi, S.A. and Siebert, X. (2023), "Multimodal Approach for Harmonized System Code Prediction", *ESANN 2023 Proceedings*, pp. 181–186, doi: 10.14428/esann/2023.ES2023-165.
- Anandarajan, M., Hill, C. and Nolan, T. (2018), *Practical Text Analytics: Maximizing the Value of Text Data*, Springer.
- Andreieva, V. and Shvai, N. (2021), "Generalization of Cross-Entropy Loss Function for Image Classification", January, doi: 10.18523/2617-7080320203-10.
- Anne, C., Chalendard, C., Fernandes, A., Rijkers, B. and Vicard, V. (2023), *Containing Tariff Evasion*, CESifo Working Paper.
- Arora, S., May, A., Zhang, J. and Ré, C. (2020), "Contextual Embeddings: When Are They Worth It?", *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, presented at the Proceedings of the 58th Annual Meeting

- of the Association for Computational Linguistics, Association for Computational Linguistics, Online, doi: 10.18653/v1/2020.acl-main.236.
- Arslan, M. and Cruz, C. (2024), “Business text classification with imbalanced data and moderately large label spaces for digital transformation”, *Applied Network Science*, SpringerOpen, Vol. 9 No. 1, pp. 1–12, doi: 10.1007/s41109-024-00623-5.
- Artetxe, M. and Schwenk, H. (2019), “Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond”, *Transactions of the Association for Computational Linguistics*, Vol. 7, pp. 597–610, doi: 10.1162/tac1_a_00288.
- de Artiñano, I.M., Depetris, F.R. and Martincus, C.V. (2023), *Automatic Product Classification in International Trade: Machine Learning and Large Language Models*, No. IDB-WP-01494, Inter-American Development Bank.
- Asakura, H. (1993), “The Harmonized System and Rules of Origin”, *Journal of World Trade*, Vol. 27 No. 4.
- Azzimonti, M., Edwards, Z., Waddell, S.R. and Wyckoff, A. (2025), “Tariffs: Estimating the Economic Impact of the 2025 Measures and Proposals”, *Federal Reserve Bank of Richmond Economic Brief*, April, available at: https://www.richmondfed.org/publications/research/economic_brief/2025/eb_25-12 (accessed 27 August 2025).
- Bahdanau, D., Cho, K. and Bengio, Y. (2016), “Neural Machine Translation by Jointly Learning to Align and Translate”, *arXiv*, 19 May, doi: 10.48550/arXiv.1409.0473.
- Bank, D., Koenigstein, N. and Giryas, R. (2021), “Autoencoders”, *arXiv*, 3 April, doi: 10.48550/arXiv.2003.05991.
- Banker, S. (2017), “Global Trade Is Powered by Artificial Intelligence”, *Forbes*.
- Barbosa, I.M.A. (2021), *Using Machine Learning to classify HS codes for Fashion Products*, Universidade do Porto, 28 June.
- Barrett, L., Fletcher, S., Kingan, R., Kumar, M., Pradhan, A. and Smey, R. (2019), “Textual Outlier Detection and Anomalies in Financial Reporting”, *2nd KDD Workshop on Anomaly Detection in Finance*, p. 6.

- Basalisco, B. (2016), "E-commerce imports into Europe: VAT and customs treatment", *E-Commerce Imports into Europe: VAT and Customs Treatment*, 4 May, available at: <https://copenhageneconomics.com/publication/e-commerce-imports-into-europe-vat-and-customs-treatment/> (accessed 10 July 2025).
- Bayer, M., Kaufhold, M.-A. and Reuter, C. (2023), "A Survey on Data Augmentation for Text Classification", *ACM Computing Surveys*, Vol. 55 No. 7, pp. 1–39, doi: 10.1145/3544558.
- Benayas, A., Sicilia, M.A. and Mora-Cantallos, M. (2024), "A Comparative Analysis of Encoder Only and Decoder Only Models in Intent Classification and Sentiment Analysis: Navigating the Trade-Offs in Model Size and Performance", *Springer Science and Business Media LLC*, 17 January, doi: 10.21203/rs.3.rs-3865391/v1.
- Bengio, Y., Courville, A. and Vincent, P. (2013), "Representation Learning: A Review and New Perspectives", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, presented at the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 35 No. 8, pp. 1798–1828, doi: 10.1109/TPAMI.2013.50.
- Bengio, Y., Ducharme, R., Vincent, P. and Janvin, C. (2003), "A Neural Probabilistic Language Model", pp. 932–938.
- Bengio, Y., Lamblin, P., Popovici, D. and Larochelle, H. (2006), "Greedy layer-wise training of deep networks", *Proceedings of the 20th International Conference on Neural Information Processing Systems*, MIT Press, Cambridge, MA, USA, pp. 153–160.
- Berahmand, K., Daneshfar, F., Salehi, E.S., Li, Y. and Xu, Y. (2024), "Autoencoders and Their Applications in Machine Learning: A Survey", *Artificial Intelligence Review*, Vol. 57 No. 2, p. 28, doi: 10.1007/s10462-023-10662-6.
- Betz, T. (2019), "Tariff Evasion and Trade Policies", *International Studies Quarterly*, Vol. 63 No. 2, pp. 380–393, doi: 10.1093/isq/sqz008.
- Bhargava, M., Vijayan, K., Anand, O. and Raina, G. (2023), "Exploration of Transfer Learning Capability of Multilingual Models for Text Classification", *Proceedings of the 2023 5th International Conference on Pattern Recognition and Intelligent Systems*, Association

- for Computing Machinery, New York, NY, USA, pp. 45–50, doi: 10.1145/3609703.3609711.
- Bhattacharjee, A., Karami, M. and Liu, H. (2022), “Text Transformations in Contrastive Self-Supervised Learning: A Review”, *arXiv*, 6 June.
- Bianco, S., Cadene, R., Celona, L. and Napoletano, P. (2018), “Benchmark Analysis of Representative Deep Neural Network Architectures”, *IEEE Access*, Vol. 6, pp. 64270–64277, doi: 10.1109/ACCESS.2018.2877890.
- Blanchard, A.E., Gao, S., Yoon, H.-J., Christian, J.B., Durbin, E.B., Wu, X.-C., Stroup, A., et al. (2022), “A Keyword-Enhanced Approach to Handle Class Imbalance in Clinical Text Classification”, *IEEE Journal of Biomedical and Health Informatics*, presented at the *IEEE Journal of Biomedical and Health Informatics*, Vol. 26 No. 6, pp. 2796–2803, doi: 10.1109/JBHI.2022.3141976.
- Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T. (2017), “Enriching Word Vectors with Subword Information”, *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135–146, doi: 10.1162/tac1_a_00051.
- Boulahia, S.Y., Amamra, A., Madi, M.R. and Daikh, S. (2021), “Early, intermediate and late fusion strategies for robust deep learning-based multimodal action recognition”, *Machine Vision and Applications*, Vol. 32 No. 6, p. 121, doi: 10.1007/s00138-021-01249-8.
- Bradley, A.P. (1997), “The use of the area under the ROC curve in the evaluation of machine learning algorithms”, *Pattern Recognition*, Vol. 30 No. 7, pp. 1145–1159, doi: 10.1016/S0031-3203(96)00142-2.
- Breiman, L.E.O. (2001), “Random Forests”, pp. 5–32.
- Breunig, M.M., Kriegel, H.-P., Ng, R.T. and Sander, J. (2000), “LOF: identifying density-based local outliers”, *SIGMOD Rec.*, Vol. 29 No. 2, pp. 93–104, doi: 10.1145/335191.335388.
- Brigstock, P.C. (2017), “Problems of interpretation and application in tariff classification; some consequential customs compliance risks and mitigation”.

- Brinkmann, A. and Bizer, C. (2021), "Improving hierarchical product classification using domain-specific language modelling.", *IEEE Data Eng. Bull.*, Vol. 44 No. 2, pp. 14–25.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., et al. (2020), "Language Models are Few-Shot Learners", *Advances in Neural Information Processing Systems*, Vol. 33, Curran Associates, Inc., pp. 1877–1901.
- Camacho-Collados, J. and Pilehvar, M.T. (2018), "On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis", in Linzen, T., Chrupała, G. and Alishahi, A. (Eds.), *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, presented at the EMNLP 2018, Association for Computational Linguistics, Brussels, Belgium, pp. 40–46, doi: 10.18653/v1/W18-5406.
- Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., St. John, R., Constant, N., et al. (2018), "Universal Sentence Encoder for English", in Blanco, E. and Lu, W. (Eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, Brussels, Belgium, pp. 169–174, doi: 10.18653/v1/D18-2029.
- Chadha, A. (2020), "Encoder vs. decoder vs. encoder-decoder models", *Distilled AI*.
- Chalapathy, R. and Chawla, S. (2019), "Deep Learning for Anomaly Detection: A Survey", *arXiv*, 23 January, doi: 10.48550/arXiv.1901.03407.
- Chalendard, C.R. (2017), "Using Internal and External Sources of Information to Reduce Customs Evasion", *SSRN Electronic Journal*, doi: 10.2139/ssrn.3120525.
- Chan, B., Ng, I. and Chung, N. (2023), "Anomaly Detection in Trade Declarations Using Deep Learning Techniques: A Risk-Assessment Approach to Identify Misclassification and Incorrect Valuation", *Statistical Journal of the IAOS*, IOS Press, Vol. 39 No. 4, pp. 825–834, doi: 10.3233/SJI-230081.
- Chandola, V., Banerjee, A. and Kumar, V. (2009), "Anomaly Detection: A Survey", *ACM Computing Surveys*, Vol. 41 No. 3, p. 15:1-15:58, doi: 10.1145/1541880.1541882.

- Che, J., Xing, Y. and Zhang, L. (2018), "A Comprehensive Solution for Deep-Learning Based Cargo Inspection to Discriminate Goods in Containers", 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), presented at the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1287–12877, doi: 10.1109/CVPRW.2018.00166.
- Che, Y., Lin, D. and Zhang, Y. (2025), "Pains or gains: Trade war, trade deficit, and tariff evasion", *Journal of International Economics*, Elsevier BV, Vol. 155, p. 104090, doi: 10.1016/j.jinteco.2025.104090.
- Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P. and Robinson, T. (2014), "One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling", *arXiv*, 4 March, doi: 10.48550/arXiv.1312.3005.
- Chen, H., Ben van Rijnsoever, Molenhuis, M., van Dijk, D., Tan, Y.H. and Rukanova, B. (2021), "The Use of Machine Learning to Identify the Correctness of HS Code for the Customs Import Declarations", 2021 IEEE 8th International Conference on Data Science and Advanced Analytics, DSAA 2021, presented at the 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA), IEEE, pp. 1–8, doi: 10.1109/DSAA53316.2021.9564203.
- Chen, L.-C. (2024), "An extended TF-IDF method for improving keyword extraction in traditional corpus-based research: An example of a climate change corpus", *Data & Knowledge Engineering*, Vol. 153, p. 102322, doi: 10.1016/j.datak.2024.102322.
- Chen, T., Kornblith, S., Norouzi, M. and Hinton, G. (2020), "A Simple Framework for Contrastive Learning of Visual Representations", *Proceedings of the 37th International Conference on Machine Learning*, presented at the International Conference on Machine Learning, PMLR, pp. 1597–1607.
- Chen, X., Bromuri, S. and van Eekelen, M. (2021), "Neural Machine Translation for Harmonized System Codes prediction", *Proceedings of 2021 6th International Conference on Machine Learning Technologies (Icmlt 2021)*, presented at the 6th

- International Conference on Machine Learning Technologies (ICMLT)*, Assoc Computing Machinery, New York, pp. 158–163, doi: 10.1145/3468891.3468915.
- Christopher Starns. (2024), “WCO Exploratory Study on the Possible Review of the Harmonised System (HS)”, 30 April, available at: <https://www.bkrconsultants.com/media-articles/review-of-the-harmonised-system> (accessed 18 July 2025).
- Chuanakrud, P., Leelanupab, T., Damrongrat, C. and Kanungsukkasem, N. (2021), “Keyword-Text Graph Representation for Short Text Classification”, 2021 13th International Conference on Information Technology and Electrical Engineering (ICITEE), presented at the 2021 13th International Conference on Information Technology and Electrical Engineering (ICITEE), pp. 24–29, doi: 10.1109/ICITEE53064.2021.9611935.
- Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. (2014), “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”, NIPS 2014 Workshop on Deep Learning, December 2014.
- Cintra, M.E., Monard, M.C. and Camargo, H. de A. (2010), “An Evaluation of Rule-Based Classification Models Induced by a Fuzzy Method and Two Classic Learning Algorithms”, 2010 Eleventh Brazilian Symposium on Neural Networks, presented at the 2010 Eleventh Brazilian Symposium on Neural Networks, pp. 188–193, doi: 10.1109/SBRN.2010.40.
- Clark, K., Luong, M.-T., Le, Q.V. and Manning, C.D. (2020), “ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators”, arXiv, 23 March, doi: 10.48550/arXiv.2003.10555.
- Cohen, J. (1988), *Statistical Power Analysis for the Behavioral Sciences*, L. Erlbaum Associates.
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., et al. (2020a), “Unsupervised Cross-lingual Representation Learning at Scale”, in Jurafsky, D., Chai, J., Schluter, N. and Tetreault, J. (Eds.), *Proceedings of the 58th*

- Annual Meeting of the Association for Computational Linguistics, presented at the ACL 2020, Association for Computational Linguistics, Online, pp. 8440–8451, doi: 10.18653/v1/2020.acl-main.747.*
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., et al. (2020b), “Unsupervised Cross-lingual Representation Learning at Scale”, in Jurafsky, D., Chai, J., Schluter, N. and Tetreault, J. (Eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, presented at the ACL 2020, Association for Computational Linguistics, Online, pp. 8440–8451, doi: 10.18653/v1/2020.acl-main.747.*
- Conneau, A. and Kiela, D. (2018), “SentEval: An Evaluation Toolkit for Universal Sentence Representations”, in Calzolari, N., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., et al. (Eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), presented at the LREC 2018, European Language Resources Association (ELRA), Miyazaki, Japan.*
- Conneau, A., Kiela, D., Schwenk, H., Barrault, L. and Bordes, A. (2017), “Supervised Learning of Universal Sentence Representations from Natural Language Inference Data”, in Palmer, M., Hwa, R. and Riedel, S. (Eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, presented at the EMNLP 2017, Association for Computational Linguistics, Copenhagen, Denmark, pp. 670–680, doi: 10.18653/v1/D17-1070.*
- Conneau, A. and Lample, G. (2019), “Cross-lingual Language Model Pretraining”, *Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc.*
- Constantine Cannon. (2020), “Constantine Cannon Client Slams the Brakes on Customs Fraud in \$8 Million Settlement Against Auto Parts Distributor Centric Parts”, Constantine Cannon, 24 July, available at: <https://constantinecannon.com/whistleblower/whistleblower-insider-blog/constantine-cannon-client-slams-brakes-on-customs-fraud/> (accessed 3 January 2025).

- Cortes, C. and Vapnik, V. (1995), "Support-Vector Networks", *Machine Learning*, Vol. 20 No. 3, pp. 273–297, doi: 10.1007/BF00994018.
- Cotacallapa, H., Saboya, N., Canas Rodrigues, P., Salas, R. and Linkolk López-Gonzales, J. (2024), "A Flat-Hierarchical Approach Based on Machine Learning Model for e-Commerce Product Classification", *IEEE Access*, Vol. 12, pp. 72730–72745, doi: 10.1109/ACCESS.2024.3400693.
- Cruz, J.C.B. and Cheng, C. (2020), "Establishing Baselines for Text Classification in Low-Resource Languages", *arXiv*, 5 May, doi: 10.48550/arXiv.2005.02068.
- Dai, H., Liu, Z., Liao, W., Huang, X., Cao, Y., Wu, Z., Zhao, L., et al. (2023), "AugGPT: Leveraging ChatGPT for Text Data Augmentation", *arXiv*, doi: 10.48550/ARXIV.2302.13007.
- Dai, W., Hwang, K. and Fan, J. (2024), "Unsupervised Anomaly Detection for Tabular Data Using Noise Evaluation", *arXiv*, 16 December, doi: 10.48550/arXiv.2412.11461.
- DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., et al. (2025), "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning", *arXiv*, 22 January, doi: 10.48550/arXiv.2501.12948.
- Demšar, J. (2006), "Statistical Comparisons of Classifiers over Multiple Data Sets", *J. Mach. Learn. Res.*, Vol. 7, pp. 1–30.
- Denny, M.J. and Spirling, A. (2018), "Text Preprocessing For Unsupervised Learning: Why It Matters, When It Misleads, And What To Do About It", *Political Analysis*, Vol. 26 No. 2, pp. 168–189, doi: 10.1017/pan.2017.44.
- Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019), "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", 2019 Conference of The North American Chapter of The Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2019), VOL. 1, presented at the Conference of the North-American-Chapter of the Association-for-Computational-Linguistics - Human Language Technologies (NAACL-HLT), Assoc Computational Linguistics-Acl, Stroudsburg, pp. 4171–4186.

- Dietterich, T.G. (1998), "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms", *Neural Computation*, Vol. 10 No. 7, pp. 1895–1923, doi: 10.1162/089976698300017197.
- Ding, L., Fan, Z. and Chen, D. (2015), "Auto-Categorization of HS Code Using Background Net Approach", *Procedia Computer Science*, Elsevier Masson SAS, Vol. 60 No. 1, pp. 1462–1471.
- Ding, Y., Korotkiy, M., Omelayenko, B., Kartseva, V., Zykov, V., Klein, M. and Schulten, E. (2002), "GoldenBullet: Automated classification of product data in e-commerce".
- Ding, Z. and Fei, M. (2013), "An Anomaly Detection Approach Based on Isolation Forest Algorithm for Streaming Data using Sliding Window", *IFAC Proceedings Volumes*, Vol. 46 No. 20, pp. 12–17, doi: 10.3182/20130902-3-CN-3020.00044.
- Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H. and Smith, N. (2020), "Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping", *arXiv*, 15 February, doi: 10.48550/arXiv.2002.06305.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., et al. (2021), "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", *arXiv*, 3 June, doi: 10.48550/arXiv.2010.11929.
- Du, S., Wu, Z., Wan, H. and Lin, Y. (2021), "HScodeNet: Combining Hierarchical Sequential and Global Spatial Information of Text for Commodity HS Code Classification", in *Karlapalem, K., Cheng, H., Ramakrishnan, N., Agrawal, R.K., Reddy, P.K., Srivastava, J. and Chakraborty, T. (Eds.), Advances in Knowledge Discovery and Data Mining, PAKDD 2021, Pt II, Vol. 12713, presented at the 25th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Springer International Publishing Ag, Cham, pp. 676–689, doi: 10.1007/978-3-030-75765-6_54.*
- Edirisinghe, L. (2017), "Customs Harmonized System Coding; Simplifying the Classification Process", *OPA Journal*, Vol. 33, pp. 61–64.

- Enigma. (2018), "US Imports - Automated Manifest System (AMS) Shipments 2018", 1 January.
- Esmaeili, F., Cassie, E., Nguyen, H.P., Plank, N., Unsworth, C. and Wang, A. (2023), "Anomaly Detection for Sensor Signals Utilizing Deep Learning Autoencoder-Based Neural Networks", *Bioengineering*, Vol. 10, doi: 10.3390/bioengineering10040405.
- Ethayarajh, K. (2019), "How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings", in Inui, K., Jiang, J., Ng, V. and Wan, X. (Eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, presented at the EMNLP-IJCNLP 2019, Association for Computational Linguistics, Hong Kong, China, pp. 55–65, doi: 10.18653/v1/D19-1006.
- European Court of Auditors. (2017), *Import Procedures: Shortcomings in the Legal Framework and an Ineffective Implementation Impact the Financial Interests of the EU*, No. 19.
- Fang, H., Wang, S., Zhou, M., Ding, J. and Xie, P. (2020), "CERT: Contrastive Self-supervised Learning for Language Understanding", *arXiv*, 18 June, doi: 10.48550/arXiv.2005.12766.
- Farhad, M., Madsen, J.B., Jetter, M. and Williams, A. (2024), "The mystery of misreported trade: A novel method to identify distorted trade figures", *The World Economy*, Wiley, Vol. 47 No. 2, pp. 748–778, doi: 10.1111/twec.13451.
- Fedotova, G. (2021), "Problems of Digital Transformation of Customs Services on Classification of Goods", *Proceedings of the 2nd International Scientific Conference on Innovations in Digital Economy*, Association for Computing Machinery, New York, NY, USA, pp. 1–10, doi: 10.1145/34444465.3444503.
- Feijo, D. de V. and Moreira, V.P. (2020), "Mono vs Multilingual Transformer-based Models: a Comparison across Several Language Tasks", *arXiv*, 19 July, doi: 10.48550/arXiv.2007.09757.

- Fernández-Delgado, M., Cernadas, E., Barro, S. and Amorim, D. (2014), "Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?", *Journal of Machine Learning Research*, Vol. 15 No. 90, pp. 3133–3181.
- Fields, J., Chovanec, K. and Madiraju, P. (2024), "A Survey of Text Classification With Transformers: How Wide? How Large? How Long? How Accurate? How Expensive? How Safe?", *IEEE Access*, Vol. 12, pp. 6518–6531, doi: 10.1109/ACCESS.2024.3349952.
- Firoozeh, N., Nazarenko, A., Alizon, F. and Daille, B. (2020), "Keyword extraction: Issues and methods", *Natural Language Engineering*, Vol. 26 No. 3, pp. 259–291, doi: 10.1017/S1351324919000457.
- Fournier, Q. and Aloise, D. (2019), "Empirical comparison between autoencoders and traditional dimensionality reduction methods", *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pp. 211–214, doi: 10.1109/AIKE.2019.00044.
- Gani, A. (2016), "The Efficiency of Customs Clearance Processes Can Matter for Trade", *International Advances in Economic Research*, Vol. 22 No. 1, pp. 109–110, doi: 10.1007/s11294-015-9556-6.
- Gao, D. (2020), "Deep Hierarchical Classification for Category Prediction in E-commerce System", in Malmasi, S., Kallumadi, S., Ueffing, N., Rokhlenko, O., Agichtein, E. and Guy, I. (Eds.), *Proceedings of the 3rd Workshop on E-Commerce and NLP, presented at the ECNLP 2020, Association for Computational Linguistics, Seattle, WA, USA*, pp. 64–68, doi: 10.18653/v1/2020.ecnlp-1.10.
- Gao, D., Yang, W., Zhou, H., Wei, Y., Hu, Y. and Wang, H. (2020), "Deep Hierarchical Classification for Category Prediction in E-commerce System", *Workshop on E-Commerce and NLP (ECNLP 3)*, presented at the 3rd International Workshop on E-Commerce and NLP (ECNLP), Assoc Computational Linguistics-Acl, Stroudsburg, pp. 64–68.

- Gao, T., Yao, X. and Chen, D. (2021), “SimCSE: Simple Contrastive Learning of Sentence Embeddings”, presented at the EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings, pp. 6894–6910.
- Garrido-Merchan, E.C., Gozalo-Brizuela, R. and Gonzalez-Carvajal, S. (2023), “Comparing BERT Against Traditional Machine Learning Models in Text Classification”, *Journal of Computational and Cognitive Engineering*, Vol. 2 No. 4, pp. 352–356, doi: 10.47852/bonviewJCCE3202838.
- Global Financial Integrity. (2021), *Trade-Related Illicit Financial Flows in 134 Developing Countries 2009-2018*.
- Godey, N., Clergerie, É. and Sagot, B. (2024), “Anisotropy is Inherent to Self-Attention in Transformers”, in Graham, Y. and Purver, M. (Eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, presented at the EACL 2024, Association for Computational Linguistics, St. Julian's, Malta, pp. 35–48.
- Goldberg, Y. (2017), *Neural Network Methods in Natural Language Processing*, Morgan & Claypool Publishers.
- Goldstein, M. and Uchida, S. (2016), “A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data”, *PLOS ONE, Public Library of Science*, Vol. 11 No. 4, p. e0152173, doi: 10.1371/journal.pone.0152173.
- Goodfellow, I., Bengio, Y. and Courville, A. (2016), *Deep Learning*, MIT Press.
- Gorokhov, O., Petrovskiy, M., Mashechkin, I. and Kazachuk, M. (2023), “Fuzzy CNN Autoencoder for Unsupervised Anomaly Detection in Log Data”, *Mathematics*, Vol. 11 No. 18, p. 3995, doi: 10.3390/math11183995.
- Grainger, A. (2024), “Customs Tariff Classification and the Use of Assistive Technologies”, *World Customs Journal, International Network of Customs Universities*, Vol. 18 No. 1, pp. 3–32, doi: 10.55596/001c.116525.

- Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., et al. (2024), "The Llama 3 Herd of Models", arXiv, 23 November, doi: 10.48550/arXiv.2407.21783.
- Grootendorst, M., Mishra, A., Matsak, A., OysterMax, Priyanshu Govil, Ogura, Y., Warmerdam, V.D., et al. (2023), "MaartenGr/KeyBERT: v0.8", Zenodo, 29 September, doi: 10.5281/ZENODO.8388690.
- Ha, T.-T., Nguyen, V.-N., Nguyen, K.-H., Nguyen, K.-A. and Than, Q.-K. (2021), "Utilizing SBERT For Finding Similar Questions in Community Question Answering", 2021 13th International Conference on Knowledge and Systems Engineering (KSE), presented at the 2021 13th International Conference on Knowledge and Systems Engineering (KSE), pp. 1–6, doi: 10.1109/KSE53942.2021.9648830.
- Han, C.-R. and Ireland, R. (2014), "Performance measurement of the KCS customs selectivity system", *Risk Management*, Vol. 16 No. 1, pp. 25–43, doi: 10.1057/rm.2014.2.
- Hanley, J.A. and McNeil, B.J. (1982), "The meaning and use of the area under a receiver operating characteristic (ROC) curve.", *Radiology*, Radiological Society of North America (RSNA), Vol. 143 No. 1, pp. 29–36, doi: 10.1148/radiology.143.1.7063747.
- Harris, Z.S. (1954), "Distributional Structure", *WORD*, Vol. 10 No. 2–3, pp. 146–162, doi: 10.1080/00437956.1954.11659520.
- Harsani, P., Suhendra, A., Wulandari, L. and Wibowo, W.C. (2020), "A Study Using Machine Learning with Ngram Model in Harmonized System Classification", *Journal of Advanced Research in Dynamical and Control Systems*, Vol. 12 No. 6 Special Issue, pp. 145–153, doi: 10.5373/JARDCS/V12SP6/SP20201018.
- Hashemzahde, B. and Abdolrazzagah-Nezhad, M. (2020), "Improving keyword extraction in multilingual texts", *International Journal of Electrical and Computer Engineering (IJECE)*, Institute of Advanced Engineering and Science, Vol. 10 No. 6, p. 5909, doi: 10.11591/ijece.v10i6.pp5909-5916.
- Hawkins, D.M. (1980), *Identification of Outliers*, Chapman and Hall.

- Hawkins, S., He, H., Williams, G. and Baxter, R. (2002), "Outlier Detection Using Replicator Neural Networks", in Kambayashi, Y., Winiwarter, W. and Arikawa, M. (Eds.), *Data Warehousing and Knowledge Discovery*, Springer, Berlin, Heidelberg, pp. 170–180, doi: 10.1007/3-540-46145-0_17.
- He, M., Wang, X., Zou, C., Dai, B. and Jin, L. (2021), "A Commodity Classification Framework Based on Machine Learning for Analysis of Trade Declaration", *Symmetry*, Vol. 13 No. 6, p. 964, doi: 10.3390/sym13060964.
- He, P., Gao, J. and Chen, W. (2023), "DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing", presented at the *International Conference on Learning Representations*, arXiv, doi: 10.48550/arXiv.2111.09543.
- Heijde, J.A. van der. (2019), *Automated Classification Tool for E-Commerce Products and Their HS Code*, Eindhoven University of Technology, Eindhoven.
- Heijmann, F., Tan, Y., Rukanova, B.D. and Veenstra, A. (2020), "The Changing Role of Customs: Customs Aligning with Supply Chain and Information Management", *World Customs Journal*, Vol. 14 No. 2.
- Hejazi, M. and Singh, Y.P. (2013), "One-Class Support Vector Machines Approach to Anomaly Detection", *Applied Artificial Intelligence*, Vol. 27 No. 5, pp. 351–366, doi: 10.1080/08839514.2013.785791.
- Henderson, M., Al-Rfou, R., Strope, B., Sung, Y., Lukacs, L., Guo, R., Kumar, S., et al. (2017), "Efficient Natural Language Response Suggestion for Smart Reply".
- Henderson, M., Casanueva, I., Mrkšić, N., Su, P.-H., Wen, T.-H. and Vulić, I. (2020), "ConveRT: Efficient and Accurate Conversational Representations from Transformers", in Cohn, T., He, Y. and Liu, Y. (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, presented at the Findings 2020, Association for Computational Linguistics, Online, pp. 2161–2174, doi: 10.18653/v1/2020.findings-emnlp.196.

- Hinton, G., Vinyals, O. and Dean, J. (2015), "Distilling the Knowledge in a Neural Network", *arXiv*, 9 March, doi: 10.48550/arXiv.1503.02531.
- Hinton, G.E. and Salakhutdinov, R.R. (2006), "Reducing the Dimensionality of Data with Neural Networks", *Science*, Vol. 313 No. 5786, pp. 504–507, doi: 10.1126/science.1127647.
- Hinton, G.E. and Zemel, R. (1993), "Autoencoders, Minimum Description Length and Helmholtz Free Energy", *Advances in Neural Information Processing Systems*, Vol. 6, Morgan-Kaufmann.
- Hochreiter, S. and Schmidhuber, J. (1997), "Long Short-Term Memory", *Neural Computation*, Vol. 9 No. 8, pp. 1735–1780, doi: 10.1162/neco.1997.9.8.1735.
- Hodge, V. and Austin, J. (2004), "A Survey of Outlier Detection Methodologies", *Artificial Intelligence Review*, Vol. 22 No. 2, pp. 85–126, doi: 10.1023/B:AIRE.0000045502.10941.a9.
- Hornik, K., Stinchcombe, M. and White, H. (1989), "Multilayer Feedforward Networks Are Universal Approximators", *Neural Networks*, Vol. 2 No. 5, pp. 359–366, doi: 10.1016/0893-6080(89)90020-8.
- Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., et al. (2021), "LoRA: Low-Rank Adaptation of Large Language Models", *arXiv*, 16 October, doi: 10.48550/arXiv.2106.09685.
- Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., et al. (2025), "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions", *ACM Trans. Inf. Syst.*, Vol. 43 No. 2, p. 42:1-42:55, doi: 10.1145/3703155.
- Huang, P., Li, L., Wu, C., Zhang, X. and Liu, Z. (2023), "A Study of Sentence-BERT Based Essay Off-topic Detection", *Proceedings of the 2023 4th International Conference on Computing, Networks and Internet of Things*, Association for Computing Machinery, New York, NY, USA, pp. 515–519, doi: 10.1145/3603781.3603871.

- Hussain, A. and Ahmed, A. (2023), "Auto-Classification of Harmonized Tariff Codes Using ChatGPT", *4th International Conference on Distributed Sensing and Intelligent Systems (ICDSIS 2023)*, Vol. 2023, presented at the 4th International Conference on Distributed Sensing and Intelligent Systems (ICDSIS 2023), pp. 262–275, doi: 10.1049/icp.2024.0497.
- Indonesian Customs. (2009), *Pemberitahuan Pabean Impor*.
- Iqbal, T. and Qureshi, S. (2023), "Reconstruction Probability-Based Anomaly Detection Using Variational Auto-Encoders", *International Journal of Computers and Applications*.
- Islam, M.Z., Liu, J., Li, J., Liu, L. and Kang, W. (2019), "A semantics aware random forest for text classification", *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Association for Computing Machinery, New York, NY, USA*, pp. 1061–1070, doi: 10.1145/3357384.3357891.
- Jahanshahi, H., Ozyegen, O., Cevik, M., Bulut, B., Yigit, D., Gonen, F.F. and Başar, A. (2021), "Text Classification for Predicting Multi-level Product Categories", *arXiv*, 2 September, doi: 10.48550/arXiv.2109.01084.
- Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D. and Makedon, F. (2021), "A Survey on Contrastive Self-Supervised Learning", *Technologies, Multidisciplinary Digital Publishing Institute*, Vol. 9 No. 1, p. 2, doi: 10.3390/technologies9010002.
- Japkowicz, N., Myers, C. and Gluck, M. (1995), "A Novelty Detection Approach to Classification", *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA*, pp. 518–523.
- Javorcik, B.S. and Narciso, G. (2017), "WTO Accession and Tariff Evasion", *Journal of Development Economics*, Vol. 125, pp. 59–71, doi: 10.1016/j.jdeveco.2016.11.001.
- Jia, H. and Liu, W. (2023), "Anomaly Detection in Images with Shared Autoencoders", *Frontiers in Neurorobotics*, Vol. 16, p. 1046867, doi: 10.3389/fnbot.2022.1046867.

- Jiang, R., Xue, Y. and Zou, D. (2023), "Interpretability-Aware Industrial Anomaly Detection Using Autoencoders", *IEEE Access*, presented at the *IEEE Access*, Vol. 11, pp. 60490–60500, doi: 10.1109/ACCESS.2023.3286548.
- Jo, T. (2018), *Text Mining: Concepts, Implementation, and Big Data Challenge*, Springer.
- Jolliffe, I.T. and Cadima, J. (2016), "Principal component analysis: a review and recent developments", *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Royal Society, Vol. 374 No. 2065, p. 20150202, doi: 10.1098/rsta.2015.0202.
- Judge, B.C. (2020), "Customs Fraud: Furniture Importers Pay", *Whistleblower Law Collaborative*, 22 April, available at: <https://www.whistleblowerllc.com/customs-fraud-settlement-furniture-importers-pay-5-2-million/> (accessed 3 January 2025).
- Kadhim, A.I. (2019), "Survey on Supervised Machine Learning Techniques for Automatic Text Classification", *Artificial Intelligence Review*, Vol. 52, pp. 273–292, doi: 10.1007/s10462-018-09677-1.
- Kannan, R., Woo, H., Aggarwal, C.C. and Park, H. (2017), "Outlier Detection for Text Data", *Proceedings of the 2017 Siam International Conference on Data Mining*, SIAM, pp. 489–497.
- Kappler, H. (2011), "Reversing the Trend: Low Cost and Low Risk Methods for Assuring Proper Duty Payments", *World Customs Journal*, Vol. 5 No. 2, doi: 10.55596/001c.92728.
- Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., et al. (2021), "Supervised Contrastive Learning", *arXiv*, 10 March, doi: 10.48550/arXiv.2004.11362.
- Kim, S., Tsai, Y.-C., Singh, K., Choi, Y., Ibok, E., Li, C.-T. and Cha, M. (2020), "DATE: Dual Attentive Tree-aware Embedding for Customs Fraud Detection", *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Association for Computing Machinery, New York, NY, USA, pp. 2880–2890, doi: 10.1145/3394486.3403339.

- Kim, S.M. (2021), *Payment Methods and Finance for International Trade*, Springer Singapore, Singapore, doi: 10.1007/978-981-15-7039-1.
- King, A. (2024), "Using Machine Translation to Augment Multilingual Classification", *arXiv*, 9 May, doi: 10.48550/arXiv.2405.05478.
- Kingma, D.P. and Welling, M. (2014), "Auto-Encoding Variational Bayes", in Bengio, Y. and LeCun, Y. (Eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.
- Kingma, D.P. and Welling, M. (2019), "An Introduction to Variational Autoencoders", *Foundations and Trends® in Machine Learning*, Vol. 12 No. 4, pp. 307–392, doi: 10.1561/22000000056.
- Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R.S., Torralba, A., Urtasun, R. and Fidler, S. (2015), "Skip-Thought Vectors", *arXiv*, 22 June, doi: 10.48550/arXiv.1506.06726.
- Kohavi, R. (1995), "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection", presented at the *International Joint Conference on Artificial Intelligence*.
- Koller, D. and Sahami, M. (1997), "Hierarchically Classifying Documents Using Very Few Words", presented at the *International Conference on Machine Learning*.
- Koto, F., Rahimi, A., Lau, J.H. and Baldwin, T. (2020), "IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP", in Scott, D., Bel, N. and Zong, C. (Eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, presented at the *COLING 2020, International Committee on Computational Linguistics, Barcelona, Spain (Online)*, pp. 757–770, doi: 10.18653/v1/2020.coling-main.66.
- Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. and Brown, D. (2019), "Text Classification Algorithms: A Survey", *Information-an International Interdisciplinary Journal*, Vol. 10 No. 4, doi: 10.3390/info10040150.
- KPMG and Reuters, T. (2016), *2016 Global Trade Management Survey*.

- Krishnan, A. and Amarthaluri, A. (2019), “Large Scale Product Categorization using Structured and Unstructured Attributes”, *arXiv*, 1 March, doi: 10.48550/arXiv.1903.04254.
- Kudo, T. and Richardson, J. (2018), “Sentencepiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing”, in Blanco, E. and Lu, W. (Eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, Brussels, Belgium, pp. 66–71, doi: 10.18653/v1/D18-2012.
- Kulagin, A., Gavrilin, Y. and Kholodov, Y. (2019), “Deep Embeddings for Brand Detection in Product Titles”, in van der Aalst, W.M.P., Batagelj, V., Ignatov, D.I., Khachay, M., Kuskova, V., Kutuzov, A., Kuznetsov, S.O., et al. (Eds.), *Analysis of Images, Social Networks and Texts*, Springer International Publishing, Cham, pp. 155–165, doi: 10.1007/978-3-030-37334-4_14.
- Kunickaite, R., Brazinskaite, A., Saltis, I. and Krilavicius, T. (2021), “Machine Learning Approaches for Customs Fraud Detection”, *IVUS*, pp. 54–63.
- Lample, G., Conneau, A., Denoyer, L. and Ranzato, M. (2018), “Unsupervised Machine Translation Using Monolingual Corpora Only”, *arXiv*, 13 April.
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P. and Soricut, R. (2020), “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations”, presented at the Eighth International Conference on Learning Representations.
- Lau, J.H. and Baldwin, T. (2016), “An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation”, *arXiv*, 19 July, doi: 10.48550/arXiv.1607.05368.
- Le, Q. and Mikolov, T. (2014), “Distributed Representations of Sentences and Documents”, *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, JMLR.org, Beijing, China, p. II-1188-II–1196.
- LeCun, Y., Bengio, Y. and Hinton, G. (2015), “Deep learning”, *Nature*, Nature Publishing Group, Vol. 521 No. 7553, pp. 436–444, doi: 10.1038/nature14539.

- LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W. and Jackel, L. (1989), "Handwritten Digit Recognition with a Back-Propagation Network", *Advances in Neural Information Processing Systems*, Vol. 2, Morgan-Kaufmann.
- Lee, Dongju, Kim, Gunwoo and Keunho Choi. (2020), "CNN-based Recommendation Model for Classifying HS Code", *Management & Information Systems Review*, Vol. 39 No. 3, pp. 1–16, doi: 10.29214/DAMIS.2020.39.3.001.
- Lee, E., Kim, S., Kim, S., Jung, S., Kim, H. and Cha, M. (2023), "Explainable Product Classification for Customs", *ACM Transactions on Intelligent Systems and Technology*, doi: 10.1145/3635158.
- Lee, E., Kim, S., Kim, S., Jung, S., Kim, H. and Cha, M. (2024), "Explainable Product Classification for Customs", *ACM Trans. Intell. Syst. Technol.*, Vol. 15 No. 2, p. 25:1-25:24, doi: 10.1145/3635158.
- Lee, E., Kim, S., Kim, S., Park, S., Cha, M., Jung, S., Yang, S., et al. (2021), "Classification of Goods Using Text Descriptions With Sentences Retrieval", *arXiv*, 2 November.
- Lehmann, E., Simonyi, A., Henkel, L. and Franke, J. (2020), "Bilingual Transfer Learning for Online Product Classification", in Zhao, H., Sondhi, P., Bach, N., Hewavitharana, S., He, Y., Si, L. and Ji, H. (Eds.), *Proceedings of Workshop on Natural Language Processing in E-Commerce*, presented at the *EcomNLP 2020*, Association for Computational Linguistics, Barcelona, Spain, pp. 21–31.
- Le-Khac, P.H., Healy, G. and Smeaton, A.F. (2020), "Contrastive Representation Learning: A Framework and Review", *IEEE Access*, Vol. 8, pp. 193907–193934, doi: 10.1109/ACCESS.2020.3031549.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., et al. (2020), "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension", presented at the *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880.

- Li, B., Hou, Y. and Che, W. (2022), "Data augmentation approaches in natural language processing: A survey", *AI Open*, Vol. 3, pp. 71–90, doi: 10.1016/j.aiopen.2022.03.001.
- Li, B., Zhou, H., He, J., Wang, M., Yang, Y. and Li, L. (2020), "On the Sentence Embeddings from Pre-trained Language Models", in Webber, B., Cohn, T., He, Y. and Liu, Y. (Eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, presented at the EMNLP 2020, Association for Computational Linguistics, Online, pp. 9119–9130, doi: 10.18653/v1/2020.emnlp-main.733.
- Li, D., Guo, H., Wang, Z., and Zhiyun Zheng. (2021), "Unsupervised Fake News Detection Based on Autoencoder", *IEEE Access*, presented at the IEEE Access, Vol. 9, doi: 10.1109/ACCESS.2021.3058809.
- Li, G. and Li, N. (2019), "Customs Classification for Cross-Border E-Commerce Based on Text-Image Adaptive Convolutional Neural Network", *Electronic Commerce Research*, Vol. 19 No. 4, pp. 779–800, doi: 10.1007/s10660-019-09334-x.
- Li, P., Pei, Y. and Li, J. (2023), "A Comprehensive Survey on Design and Application of Autoencoder in Deep Learning", *Applied Soft Computing*, Vol. 138, p. 110176, doi: 10.1016/j.asoc.2023.110176.
- Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., Yu, P.S., et al. (2022), "A Survey on Text Classification: From Traditional to Deep Learning", *ACM Transactions on Intelligent Systems and Technology*, Vol. 13 No. 2, pp. 1–41, doi: 10.1145/3495162.
- Li, X. (2024), "Comparative analysis and prospect of RNN and Transformer", *Applied and Computational Engineering*, Vol. 75 No. 1, pp. 178–184, doi: 10.54254/2755-2721/75/20240535.
- Li, X., Lei, Y. and Ji, S. (2022), "BERT- and BiLSTM-Based Sentiment Analysis of Online Chinese Buzzwords", *Future Internet*, Multidisciplinary Digital Publishing Institute, Vol. 14 No. 11, p. 332, doi: 10.3390/fi14110332.

- Li, X., Li, Z., Sheng, J. and Slamu, W. (2020), “Low-Resource Text Classification via Cross-lingual Language Model Fine-tuning”, in Sun, M., Li, S., Zhang, Y. and Liu, Y. (Eds.), *Proceedings of the 19th Chinese National Conference on Computational Linguistics*, presented at the CCL 2020, Chinese Information Processing Society of China, Haikou, China, pp. 994–1005.
- Li, Y., Li, J., Xiao, Z., Yang, T., Nian, Y., Hu, X. and Zhao, Y. (2024), “NLP-ADBench: NLP Anomaly Detection Benchmark”, arXiv, 6 December, doi: 10.48550/arXiv.2412.04784.
- Li, Y., Ma, W., Chen, C., Zhang, M., Liu, Y., Ma, S. and Yang, Y. (2022), “A Survey on Dropout Methods and Experimental Verification in Recommendation”, arXiv, 14 May, doi: 10.48550/arXiv.2204.02027.
- Li, Z., Li, X., Liu, Y., Xie, H., Li, J., Wang, F., Li, Q., et al. (2023), “Label Supervised LLaMA Finetuning”, arXiv, 2 October, doi: 10.48550/arXiv.2310.01208.
- Liang, Y., Cao, R., Zheng, J., Ren, J. and Gao, L. (2021), “Learning to Remove: Towards Isotropic Pre-trained BERT Embedding”, in Farkaš, I., Masulli, P., Otte, S. and Wermter, S. (Eds.), *Artificial Neural Networks and Machine Learning – ICANN 2021*, Springer International Publishing, Cham, pp. 448–459, doi: 10.1007/978-3-030-86383-8_36.
- Liao, D. (2021), “Sentence Embeddings using Supervised Contrastive Learning”, arXiv, 8 June.
- Liao, M., Huang, L., Zhang, J., Song, L. and Li, B. (2024), “Enhanced HS Code Classification for Import and Export Goods via Multiscale Attention and ERNIE-BiLSTM”, *Applied Sciences (Switzerland)*, Vol. 14 No. 22, doi: 10.3390/app142210267.
- de Lima, R.R., Fernandes, A.M.R., Bombasar, J.R., da Silva, B.A., Crocker, P. and Leithardt, V.R.Q. (2022), “An Empirical Comparison of Portuguese and Multilingual BERT Models for Auto-Classification of NCM Codes in International Trade”, *Big Data and Cognitive Computing*, Vol. 6 No. 1, doi: 10.3390/bdcc6010008.
- Lin, T., Wang, Y., Liu, X. and Qiu, X. (2021), “A Survey of Transformers”, arXiv, 15 June, doi: 10.48550/arXiv.2106.04554.

- Liu, C., Zhu, W., Zhang, X. and Zhai, Q. (2023), "Sentence part-enhanced BERT with respect to downstream tasks", *Complex & Intelligent Systems*, Springer Heidelberg, Heidelberg, Vol. 9 No. 1, pp. 463–474, doi: 10.1007/s40747-022-00819-1.
- Liu, F.T., Ting, K.M. and Zhou, Z.H. (2008), "Isolation forest", *Proceedings - IEEE International Conference on Data Mining, ICDM, IEEE*, pp. 413–422, doi: 10.1109/ICDM.2008.17.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., et al. (2019), "RoBERTa: A Robustly Optimized BERT Pretraining Approach", *arXiv*, 26 July, doi: 10.48550/arXiv.1907.11692.
- Lohmann, J. (2011), "Do language barriers affect trade?", *Economics Letters*, Elsevier BV, Vol. 110 No. 2, pp. 159–162, doi: 10.1016/j.econlet.2010.10.023.
- Lombardi, S., Gorgoglione, M. and Panniello, U. (2013), "The effect of context on misclassification costs in e-commerce applications", *Expert Systems with Applications*, Vol. 40 No. 13, pp. 5219–5227, doi: 10.1016/j.eswa.2013.03.009.
- Loshchilov, I. and Hutter, F. (2017), "Decoupled Weight Decay Regularization", presented at the *International Conference on Learning Representations*.
- Lukasik, M., Bhojanapalli, S., Menon, A.K. and Kumar, S. (2020), "Does label smoothing mitigate label noise?", *arXiv*, 5 March, doi: 10.48550/arXiv.2003.02819.
- Luong, T., Pham, H. and Manning, C.D. (2015), "Effective Approaches to Attention-based Neural Machine Translation", in Màrquez, L., Callison-Burch, C. and Su, J. (Eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, presented at the EMNLP 2015, Association for Computational Linguistics, Lisbon, Portugal, pp. 1412–1421, doi: 10.18653/v1/D15-1166.
- Luppès, J. (2019), *Classifying Short Text for the Harmonized System with Convolutional Neural Networks*, Radboud University, 23 August.
- Lushi, I. and Gashi, R. (2016), "Potential Risks to Avoidance of Customs Duties: Research in Kosovo", *European Journal of Sustainable Development*, Vol. 5 No. 2, pp. 67–74, doi: 10.14207/ejsd.2016.v5n2p67.

- Lux, M. and Matt, C. (2021), "Classification of Goods: What are the Hurdles and Pitfalls in the Use of Automation or IT Support?", *Global Trade and Customs Journal*, Vol. 16 No. 6, pp. 237–247.
- Maaten, L. van der and Hinton, G. (2008), "Visualizing Data using t-SNE", *Journal of Machine Learning Research*, Vol. 9 No. 86, pp. 2579–2605.
- Mahapatra, A., Srivastava, N. and Srivastava, J. (2012), "Contextual Anomaly Detection in Text Data", *Algorithms*, Vol. 5 No. 4, pp. 469–489, doi: 10.3390/a5040469.
- Mai, T.-D., Hoang, K., Baigutanova, A., Alina, G. and Kim, S. (2021), "Customs Fraud Detection in the Presence of Concept Drift", *2021 International Conference on Data Mining Workshops (ICDMW)*, presented at the 2021 International Conference on Data Mining Workshops (ICDMW), pp. 370–379, doi: 10.1109/ICDMW53433.2021.00052.
- Mangrulkar, S., M S, A. and Sembium, V. (2022), "Multilingual Semantic Sourcing using Product Images for Cross-lingual Alignment", *Companion Proceedings of the Web Conference 2022*, Association for Computing Machinery, New York, NY, USA, pp. 41–51, doi: 10.1145/3487553.3524204.
- Marra de Artiñano, I., Riottini Depetris, F. and Volpe Martincus, C. (2023), *Automatic Product Classification in International Trade: Machine Learning and Large Language Models*, Working Paper No. IDB-WP-01494, Inter-American Development Bank, doi: 10.18235/0005012.
- Mars, M. (2022), "From Word Embeddings to Pre-Trained Language Models: A State-of-the-Art Walkthrough", *Applied Sciences*, Multidisciplinary Digital Publishing Institute, Vol. 12 No. 17, p. 8805, doi: 10.3390/app12178805.
- Masakuna, J.F., Nkashama, Dj.K., Soltani, A., Frappier, M., Tardif, P.-M. and Kabanza, F. (2024), "Impact of Inaccurate Contamination Ratio on Robust Unsupervised Anomaly Detection", *arXiv*, 14 August, doi: 10.48550/arXiv.2408.07718.
- Matsudaira, T. and Koh, J. (2022), "Customs administration and digitalization", *Customs Matters*, International Monetary Fund.

- Mengistu, A.T., Molla, K.G. and Mascagni, G. (2019), *Tax Evasion and Missing Imports: Evidence From Transaction-Level Data*, report, The Institute of Development Studies and Partner Organisations.
- Michelucci, U. (2022), "An Introduction to Autoencoders", arXiv, 11 January.
- Miguel-Diez, A., Campazas-Vega, A., Álvarez-Aparicio, C., Esteban-Costales, G. and Guerrero-Higueras, Á.M. (2025), "A Systematic Literature Review of Unsupervised Learning Algorithms for Anomalous Traffic Detection Based on Flows", arXiv, 11 March, doi: 10.48550/arXiv.2503.08293.
- Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013), "Efficient Estimation of Word Representations in Vector Space", presented at the 1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings.
- Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M. and Gao, J. (2021), "Deep Learning Based Text Classification: A Comprehensive Review", arXiv, 4 January, doi: 10.48550/arXiv.2004.03705.
- Mishra, M.K. and Viradiya, J. (2019), "Survey of Sentence Embedding Methods", Unpublished, doi: 10.13140/RG.2.2.21861.45289.
- Mitkov, R. (Ed.). (2005), *The Oxford Handbook of Computational Linguistics*, Oxford University Press, USA, Oxford.
- Mohammadshahi, A., Lebrete, R. and Aberer, K. (2019), "Aligning Multilingual Word Embeddings for Cross-Modal Retrieval Task", in Mogadala, A., Klakow, D., Pezzelle, S. and Moens, M.-F. (Eds.), *Proceedings of the Beyond Vision and LANguage: inTEgrating Real-World kNowledge (LANTERN)*, Association for Computational Linguistics, Hong Kong, China, pp. 11–17, doi: 10.18653/v1/D19-6402.
- Mohebbian, M.R., Wahid, K.A. and Babyn, P. (2021), "Stack of Discriminative Autoencoders for Multiclass Anomaly Detection in Endoscopy Images", arXiv, 19 August.
- Mokoatle, M., Marivate, V., Mapiye, D., Bornman, R. and Hayes, Vanessa.M. (2023), "A Review and Comparative Study of Cancer Detection Using Machine Learning:

- SBERT and SimCSE Application*”, *BMC Bioinformatics*, Vol. 24 No. 1, p. 112, doi: 10.1186/s12859-023-05235-x.
- Moukafih, Y., Ghanem, A., Abidi, K., Sbihi, N., Ghogho, M. and Smaili, K. (2022), “*SimSCL: A Simple Fully-Supervised Contrastive Learning Framework for Text Representation*”, in Long, G., Yu, X. and Wang, S. (Eds.), *AI 2021: Advances in Artificial Intelligence*, Springer International Publishing, Cham, pp. 728–738, doi: 10.1007/978-3-030-97546-3_59.
- Nair, A.R., Singh, R.P., Gupta, D. and Kumar, P. (2024), “*Evaluating the Impact of Text Data Augmentation on Text Classification Tasks using DistilBERT*”, *Procedia Computer Science*, Vol. 235, pp. 102–111, doi: 10.1016/j.procs.2024.04.013.
- Naseem, U., Razzak, I. and Eklund, P.W. (2021), “*A Survey of Pre-Processing Techniques to Improve Short-Text Quality: A Case Study on Hate Speech Detection on Twitter*”, *Multimedia Tools and Applications*, Vol. 80 No. 28, pp. 35239–35266, doi: 10.1007/s11042-020-10082-6.
- Naseem, U., Razzak, I., Khan, S.K. and Prasad, M. (2021), “*A Comprehensive Survey on Word Representation Models: From Classical to State-of-the-Art Word Representation Language Models*”, *ACM Transactions on Asian and Low-Resource Language Information Processing*, Vol. 20 No. 5, pp. 1–35, doi: 10.1145/3434237.
- Navasardyan, Z. (2024), “*Interpretable and Generalizable HTS Code Classification Framework*”, *Economics, Finance and Accounting*, Vol. 1 No. 13, pp. 140–140, doi: 10.59503/29538009-2024.1.13-140.
- Nawaz, A., Khan, S.S. and Ahmad, A. (2024), “*Ensemble of Autoencoders for Anomaly Detection in Biomedical Data: A Narrative Review*”, *IEEE Access*, presented at the *IEEE Access*, Vol. 12, pp. 17273–17289, doi: 10.1109/ACCESS.2024.3360691.
- Nolle, T., Luetzgen, S., Seeliger, A. and Mühlhäuser, M. (2018), “*Analyzing Business Process Anomalies Using Autoencoders*”, *Machine Learning*, Vol. 107 No. 11, pp. 1875–1893, doi: 10.1007/s10994-018-5702-8.

- Nomoto, T. (2022), "Keyword Extraction: A Modern Perspective", *SN Computer Science*, Vol. 4 No. 1, p. 92, doi: 10.1007/s42979-022-01481-7.
- Novith, D.C. (2024), "Harmonized System Code Recommendation: A Multi-Class Classification Model", *Jurnal BPPK: Badan Pendidikan Dan Pelatihan Keuangan*, Vol. 17 No. 3, pp. 1–11.
- Novoa-Paradela, D., Fontenla-Romero, O. and Guijarro-Berdiñas, B. (2024), "Explained anomaly detection in text reviews: Can subjective scenarios be correctly evaluated?", *Engineering Applications of Artificial Intelligence*, Vol. 133, p. 108065, doi: 10.1016/j.engappai.2024.108065.
- Office of The Auditor General of Canada. (2017), *2017 Spring Reports of the Auditor General of Canada to the Parliament of Canada*.
- Onan, A., Korukoğlu, S. and Bulut, H. (2016), "Ensemble of keyword extraction methods and classifiers in text classification", *Expert Systems with Applications*, Vol. 57, pp. 232–247, doi: 10.1016/j.eswa.2016.03.045.
- Oord, A. van den, Li, Y. and Vinyals, O. (2019), "Representation Learning with Contrastive Predictive Coding", *arXiv*, 22 January, doi: 10.48550/arXiv.1807.03748.
- Ozyegen, O., Jahanshahi, H., Cevik, M., Bulut, B., Yigit, D., Gonen, F.F. and Başar, A. (2022), "Classifying multi-level product categories using dynamic masking and transformer models", *Journal of Data, Information and Management*, Vol. 4 No. 1, pp. 71–85, doi: 10.1007/s42488-022-00066-6.
- Pain, K. (2021), *Harmonized System Code Classification Using Transfer Learning With Pre-Trained Weights*, Dalhousie University, Halifax.
- Pan, S.J. and Yang, Q. (2010), "A Survey on Transfer Learning", *IEEE Transactions on Knowledge and Data Engineering*, presented at the *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22 No. 10, pp. 1345–1359, doi: 10.1109/TKDE.2009.191.

- Pang, G., Shen, C., Cao, L. and Hengel, A.V.D. (2021), "Deep Learning for Anomaly Detection: A Review", *ACM Comput. Surv.*, Vol. 54 No. 2, p. 38:1-38:38, doi: 10.1145/3439950.
- Pang, G., Shen, C., Cao, L. and Hengel, A.V.D. (2022), "Deep Learning for Anomaly Detection: A Review", *ACM Computing Surveys*, Vol. 54 No. 2, pp. 1–38, doi: 10.1145/3439950.
- Pang, T., Xu, K., Dong, Y., Du, C., Chen, N. and Zhu, J. (2020), "Rethinking Softmax Cross-Entropy Loss for Adversarial Robustness", *8th International Conference on Learning Representations, ICLR 2020*, Vol. 10, pp. 1–19.
- Paramartha, I.G.Y., Ardiyanto, I. and Hidayat, R. (2021), "Developing Machine Learning Framework to Classify Harmonized System Code. Case Study: Indonesian Customs", *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, presented at the 2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT), pp. 254–259, doi: 10.1109/EIConCIT50028.2021.9431888.
- Park, M., Choi, S., Choi, C., Kim, J.-S. and Sohn, J. (2024), "Improving Multi-lingual Alignment Through Soft Contrastive Learning", in Cao, Y. (Trista), Papadimitriou, I., Ovalle, A., Zampieri, M., Ferraro, F. and Swayamdipta, S. (Eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, Association for Computational Linguistics, Mexico City, Mexico, pp. 138–145, doi: 10.18653/v1/2024.naacl-srw.16.
- Peeters, R. and Bizer, C. (2022), "Cross-Language Learning for Product Matching", *Companion Proceedings of the Web Conference 2022*, presented at the WWW '22: The ACM Web Conference 2022, ACM, Virtual Event, Lyon France, pp. 236–238, doi: 10.1145/3487553.3524234.
- Pennington, J., Socher, R. and Manning, C. (2014), "GloVe: Global Vectors for Word Representation", in Moschitti, A., Pang, B. and Daelemans, W. (Eds.), *Proceedings of*

- the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, presented at the EMNLP 2014, Association for Computational Linguistics, Doha, Qatar, pp. 1532–1543, doi: 10.3115/v1/D14-1162.
- Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L. (2018), “Deep Contextualized Word Representations”, in Walker, M., Ji, H. and Stent, A. (Eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, presented at the NAACL-HLT 2018, Association for Computational Linguistics, New Orleans, Louisiana, pp. 2227–2237, doi: 10.18653/v1/N18-1202.
- Pires, T., Schlinger, E. and Garrette, D. (2019), “How Multilingual is Multilingual BERT?”, in Korhonen, A., Traum, D. and Màrquez, L. (Eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, presented at the ACL 2019, Association for Computational Linguistics, Florence, Italy, pp. 4996–5001, doi: 10.18653/v1/P19-1493.
- Prost, E.R.D. and Scattolo, G. (2024), “Inspections in customs: a case study of the role of tariff dispersion on red lane classification in argentina”, *World Customs Journal, International Network of Customs Universities*, Vol. 18 No. 1, pp. 33–49.
- Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N. and Huang, X. (2020), “Pre-trained Models for Natural Language Processing: A Survey”, *Science China Technological Sciences*, Vol. 63 No. 10, pp. 1872–1897, doi: 10.1007/s11431-020-1647-3.
- Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., et al. (2021), “Learning Transferable Visual Models From Natural Language Supervision”, in Meila, M. and Zhang, T. (Eds.), *International Conference on Machine Learning*, Vol. 139, Vol. 139, presented at the International Conference on Machine Learning (ICML), Jmlr-Journal Machine Learning Research, San Diego.
- Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I. (2018), “Improving Language Understanding by Generative Pre-Training”.

- Raihan, M.A., Goli, N. and Aamodt, T.M. (2019), “Modeling Deep Learning Accelerator Enabled GPUs”, 2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), presented at the 2019 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp. 79–92, doi: 10.1109/ISPASS.2019.00016.
- Rajaei, S. and Pilehvar, M.T. (2021), “How Does Fine-tuning Affect the Geometry of Embedding Space: A Case Study on Isotropy”, in Moens, M.-F., Huang, X., Specia, L. and Yih, S.W. (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, presented at the Findings 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, pp. 3042–3049, doi: 10.18653/v1/2021.findings-emnlp.261.
- Ramesh Kashyap, A., Nguyen, T.-T., Schlegel, V., Winkler, S., Ng, S.-K. and Poria, S. (2024), “A Comprehensive Survey of Sentence Representations: From the BERT Epoch to the CHATGPT Era and Beyond”, in Graham, Y. and Purver, M. (Eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, presented at the EACL 2024, Association for Computational Linguistics, St. Julian’s, Malta, pp. 1738–1751.
- Ranjan, R., Patel, V.M. and Chellappa, R. (2017), “HyperFace: A Deep Multi-Task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 41 No. 1, pp. 121–135, doi: 10.1109/TPAMI.2017.2781233.
- Rao, N. (2020), “Multilingual shopping systems”, Amazon Science, Search and information retrieval, , 8 January, available at: <https://www.amazon.science/blog/multilingual-shopping-systems> (accessed 18 July 2025).
- Raschka, S., Liu, Y. (Hayden), Mirjalili, V. and Dzhuigakov, D. (2022), *Machine Learning with Pytorch and Scikit-Learn*, Packt Publishing Ltd.
- Reimers, N. and Gurevych, I. (2019), “Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks”, *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical*

- Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp. 3982–3992, doi: 10.18653/v1/d19-1410.
- Rezende, P.M., Xavier, J.S., Ascher, D.B., Fernandes, G.R. and Pires, D.E.V. (2022), “Evaluating hierarchical machine learning approaches to classify biological databases”, *Briefings in Bioinformatics*, Vol. 23 No. 4, p. bbac216, doi: 10.1093/bib/bbac216.
- Ribeiro, M.T., Singh, S. and Guestrin, C. (2016), ““Why Should I Trust You?’: Explaining the Predictions of Any Classifier”, *arXiv*, 9 August, doi: 10.48550/arXiv.1602.04938.
- Robertson, S. (2004), “Understanding inverse document frequency: on theoretical arguments for IDF”, *Journal of Documentation*, Emerald Group Publishing Limited, Vol. 60 No. 5, pp. 503–520, doi: 10.1108/00220410410560582.
- Roman, N.T., Constantino, E.R., Ribeiro, H., Filho, J.J., Lanna, A., Goldenstein, S.K. and Wainer, J. (2006), “Carancho-a decision support system for customs”, *Practical Data Mining: Applications, Experiences and Challenges*, p. 100.
- Ruder, D. (2020), *Application of Machine Learning for Automated HS-6 Code Assignment*, Master Thesis, Tallinn University of Technology, Tallinn.
- Ruder, S., Vulić, I. and Søgaard, A. (2019), “A Survey Of Cross-lingual Word Embedding Models”, *Journal of Artificial Intelligence Research*, Vol. 65, pp. 569–631, doi: 10.1613/jair.1.11640.
- Rudman, W. and Eickhoff, C. (2024), “Stable Anisotropic Regularization”, *arXiv*, 4 April, doi: 10.48550/arXiv.2305.19358.
- Ruff, L., Kauffmann, J.R., Vandermeulen, R.A., Montavon, G., Samek, W., Kloft, M., Dietterich, T.G., et al. (2021), “A Unifying Review of Deep and Shallow Anomaly Detection”, *Proceedings of the IEEE*, Vol. 109 No. 5, pp. 756–795, doi: 10.1109/JPROC.2021.3052449.
- Rukanova, B., Tan, Y.-H., Slegt, M., Molenhuis, M., van Rijnsoever, B., Migeotte, J., Labare, M.L.M., et al. (2021), “Identifying the Value of Data Analytics in the Context of

- Government Supervision: Insights from the Customs Domain*", *Government Information Quarterly*, Vol. 38 No. 1, p. 101496, doi: 10.1016/j.giq.2020.101496.
- Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986), "Learning Representations by Back-Propagating Errors", *Nature*, Nature Publishing Group, Vol. 323 No. 6088, pp. 533–536, doi: 10.1038/323533a0.
- Rybak, N. and Hassall, M. (2021), "Deep Learning Unsupervised Text-Based Detection of Anomalies in U.S. Chemical Safety and Hazard Investigation Board Reports", 2021 *International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, presented at the 2021 *International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pp. 1–7, doi: 10.1109/ICECCME52200.2021.9590834.
- Ryzhova, A. and Sochenkov, I. (2019), "Deep Learning for Customs Classification of Goods Based on Their Textual Descriptions Analysis", 2019 *Ivannikov Ispras Open Conference (ISPRAS)*, presented at the 2019 *Ivannikov Ispras Open Conference (ISPRAS)*, pp. 55–59, doi: 10.1109/ISPRAS47671.2019.00014.
- Sakurada, M. and Yairi, T. (2014), "Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction", *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, Association for Computing Machinery, New York, NY, USA, pp. 4–11, doi: 10.1145/2689746.2689747.
- Salton, G. and Buckley, C. (1988), "Term-Weighting Approaches in Automatic Text Retrieval", *Information Processing & Management*, Vol. 24 No. 5, pp. 513–523, doi: 10.1016/0306-4573(88)90021-0.
- Sanh, V., Debut, L., Chaumond, J. and Wolf, T. (2020), "Distilbert, a Distilled Version of Bert: Smaller, Faster, Cheaper and Lighter", *arXiv*, 29 February, doi: 10.48550/arXiv.1910.01108.
- Sarker, I.H. (2021), "Machine Learning: Algorithms, Real-World Applications and Research Directions", *SN Computer Science*, Springer Science and Business Media LLC, Vol. 2 No. 3, doi: 10.1007/s42979-021-00592-x.

- Schlegel, V., Nenadic, G. and Batista-Navarro, R. (2021), "Semantics Altering Modifications for Evaluating Comprehension in Machine Reading", *arXiv*, 15 June, doi: 10.48550/arXiv.2012.04056.
- Schuster, M. and Nakajima, K. (2012), "Japanese and Korean voice search", 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), presented at the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5149–5152, doi: 10.1109/ICASSP.2012.6289079.
- Schuster, M. and Paliwal, K.K. (1997), "Bidirectional Recurrent Neural Networks", *IEEE Transactions on Signal Processing*, Vol. 45 No. 11, pp. 2673–2681, doi: 10.1109/78.650093.
- Schuster, T., Ram, O., Barzilay, R. and Globerson, A. (2019), "Cross-Lingual Alignment of Contextual Word Embeddings, with Applications to Zero-shot Dependency Parsing", in Burstein, J., Doran, C. and Solorio, T. (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, presented at the NAACL-HLT 2019, Association for Computational Linguistics, Minneapolis, Minnesota, pp. 1599–1613, doi: 10.18653/v1/N19-1162.
- Sennrich, R., Haddow, B. and Birch, A. (2016), "Neural Machine Translation of Rare Words with Subword Units", *arXiv*, 10 June, doi: 10.48550/arXiv.1508.07909.
- Seyfioğlu, M.S., Özbayoğlu, A.M. and Gürbüz, S.Z. (2018), "Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 54 No. 4, pp. 1709–1723, doi: 10.1109/TAES.2018.2799758.
- Shakil, M.H. and Tasnia, M. (2022), "Artificial Intelligence and Tax Administration in Asia and the Pacific", *Taxation in the Digital Economy*, Routledge.
- Sharma, P. and Parwekar, P. (2023), "Comparing Ensemble Techniques for Bilingual Multiclass Classification of Online Reviews", Vol. 681, presented at the *Lecture Notes in Networks and Systems*, pp. 275–284, doi: 10.1007/978-981-99-1909-3_25.

- Shubham, Arya, A., Roy, S. and Jonnala, S. (2023), "An Ensemble-Based Approach for Assigning Text to Correct Harmonized System Code", 2023 International Conference on Artificial Intelligence and Smart Communication (AISC), presented at the 2023 International Conference on Artificial Intelligence and Smart Communication (AISC), pp. 35–41, doi: 10.1109/AISC56616.2023.10085512.
- Siddiqi, S. and Sharan, A. (2015), "Keyword and Keyphrase Extraction Techniques: A Literature Review", *International Journal of Computer Applications*, Vol. 109 No. 2, pp. 18–23.
- Siino, M., Di Nuovo, E., Tinnirello, I., La Cascia, M., and others. (2021), "Detection of Hate Speech Spreaders Using Convolutional Neural Networks", *CLEF (Working Notes)*, pp. 2126–2136.
- Siino, M., Tinnirello, I. and La Cascia, M. (2024), "Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers", *Information Systems*, Vol. 121, p. 102342, doi: 10.1016/j.is.2023.102342.
- Silla, C.N. and Freitas, A.A. (2011), "A Survey of Hierarchical Classification Across Different Application Domains", *Data Mining and Knowledge Discovery*, Vol. 22 No. 1, pp. 31–72, doi: 10.1007/s10618-010-0175-9.
- Singh, A.K. and Sahu, R. (2004), "Decision Support System for HS Classification of Commodities".
- Souza, F., Nogueira, R. and Lotufo, R. (2020), "BERTimbau: Pretrained BERT Models for Brazilian Portuguese", in Cerri, R. and Prati, R.C. (Eds.), *Intelligent Systems*, Springer International Publishing, Cham, pp. 403–417, doi: 10.1007/978-3-030-61377-8_28.
- Sparck Jones, K. (1972), "A Statistical Interpretation of Term Specificity and Its Application in Retrieval", *Journal of Documentation*, MCB UP Ltd, Vol. 28 No. 1, pp. 11–21.
- Spichakova, M. and Haav, H.-M. (2020a), "Application of Machine Learning for Assessment of HS Code Correctness", *Baltic Journal of Modern Computing*, presented at the 14th

- International Baltic Conference on Databases and Information Systems (Baltic DB & IS)*, Univ Latvia, Riga, Vol. 8 No. 4, pp. 698–718, doi: 10.22364/bjmc.2020.8.4.13.
- Spichakova, M. and Haav, H.-M. (2020b), “Using Machine Learning for Automated Assessment of Misclassification of Goods for Fraud Detection”, in Robal, T., Haav, H.-M., Penjam, J. and Matulevičius, R. (Eds.), *Databases and Information Systems*, Springer International Publishing, Cham, pp. 144–158, doi: 10.1007/978-3-030-57672-1_12.
- Stassin, S., Amel, O., Mahmoudi, S.A. and Siebert, X. (2023), “Similarity versus Supervision: Best Approaches for HS Code Prediction”, *ESANN 2023 Proceedings*, presented at the ESANN 2023 - European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Ciaco - i6doc.com, Bruges (Belgium) and online, pp. 175–180, doi: 10.14428/esann/2023.ES2023-163.
- Su, J., Cao, J., Liu, W. and Ou, Y. (2021), “Whitening Sentence Representations for Better Semantics and Faster Retrieval”, *arXiv*, 28 March.
- Subakti, A., Murfi, H. and Hariadi, N. (2022), “The Performance of Bert as Data Representation of Text Clustering”, *Journal of Big Data*, Vol. 9 No. 1, p. 15, doi: 10.1186/s40537-022-00564-9.
- Sun, A. and Lim, E.-P. (2001), “Hierarchical text classification and evaluation”, *Proceedings 2001 IEEE International Conference on Data Mining*, presented at the 2001 IEEE International Conference on Data Mining, pp. 521–528, doi: 10.1109/ICDM.2001.989560.
- Sun, C., Qiu, X., Xu, Y. and Huang, X. (2019), “How to Fine-Tune BERT for Text Classification?”, in Sun, M., Huang, X., Ji, H., Liu, Z. and Liu, Y. (Eds.), *Chinese Computational Linguistics*, Vol. 11856, Springer International Publishing, Cham, pp. 194–206, doi: 10.1007/978-3-030-32381-3_16.
- Sun, C., Qiu, X., Xu, Y. and Huang, X. (2020), “How to Fine-Tune BERT for Text Classification?”, *arXiv*, 5 February.

- Sutskever, I., Vinyals, O. and Le, Q.V. (2014), "Sequence to Sequence Learning with Neural Networks", *Advances in Neural Information Processing Systems*, Vol. 27, Curran Associates, Inc.
- Szymański, J. (2014), "Comparative Analysis of Text Representation Methods Using Classification", *Cybernetics and Systems*, Taylor & Francis, Vol. 45 No. 2, pp. 180–199, doi: 10.1080/01969722.2014.874828.
- Taiye, M.A. (2019), *An Enhanced Sequential Exception Technique for Semantic-Based Text Anomaly Detection*, PhD Theses, Universiti Utara Malaysia, 2 May.
- Tang, Z., Dai, D., Chen, Z. and Chen, T. (2022), "Short Text Classification Combining Keywords and Knowledge", *2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, presented at the 2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE), pp. 662–665, doi: 10.1109/ICCECE54139.2022.9712673.
- Tec Yan Yap. (2020), "Text Anomaly Detection with ARAE-AnoGAN", *Honors Projects*.
- Tenzer, H., Terjesen, S. and Harzing, A.-W. (2017), "Language in International Business: A Review and Agenda for Future Research", *Management International Review*, Springer Science and Business Media LLC, Vol. 57 No. 6, pp. 815–854, doi: 10.1007/s11575-017-0319-x.
- Thakur, N., Reimers, N., Daxenberger, J. and Gurevych, I. (2021), "Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks", *2021 CONFERENCE OF THE NORTH AMERICAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: HUMAN LANGUAGE TECHNOLOGIES (NAACL-HLT 2021)*, presented at the Conference of the North-American-Chapter of the Association-for-Computational-Linguistics - Human Language Technologies (NAACL-HLT), Assoc Computational Linguistics-Acl, Stroudsburg, pp. 296–310.
- Thakur, N., Reimers, N., Rücklé, A., Srivastava, A. and Gurevych, I. (2021), "BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models",

Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2).

- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., et al. (2023), "LLaMA: Open and Efficient Foundation Language Models", *arXiv*, 27 February, doi: 10.48550/arXiv.2302.13971.
- Triepels, R., Daniels, H. and Feelders, A. (2018), "Data-Driven Fraud Detection in International Shipping", *Expert Systems with Applications, Elsevier Ltd*, Vol. 99, pp. 193–202, doi: 10.1016/j.eswa.2018.01.007.
- Turhan, B., Akar, G.B., Turhan, C. and Yukse, C. (2015), "Visual and Textual Feature Fusion for Automatic Customs Tariff Classification", *2015 IEEE International Conference on Information Reuse and Integration, presented at the 2015 IEEE International Conference on Information Reuse and Integration*, pp. 76–81, doi: 10.1109/IRI.2015.22.
- Uysal, A.K. and Gunal, S. (2014), "The Impact of Preprocessing on Text Classification", *Information Processing & Management*, Vol. 50 No. 1, pp. 104–112, doi: 10.1016/j.ipm.2013.08.006.
- Valeriani, L., Doimo, D., Cuturello, F., Laio, A., Ansuini, A. and Cazzaniga, A. (2023), "The geometry of hidden representations of large transformer models", *Proceedings of the 37th International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA*, pp. 51234–51252.
- Vanhoeyveld, J., Martens, D. and Peeters, B. (2020), "Customs Fraud Detection: Assessing the Value of Behavioural and High-Cardinality Data Under the Imbalanced Learning Issue", *Pattern Analysis and Applications, Springer London*, Vol. 23 No. 3, pp. 1457–1477, doi: 10.1007/s10044-019-00852-w.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. ukasz, et al. (2017), "Attention is All you Need", *Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc.*

- Vig, J. (2019), "A Multiscale Visualization of Attention in the Transformer Model", in Costa-jussà, M.R. and Alfonseca, E. (Eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Association for Computational Linguistics, Florence, Italy, pp. 37–42, doi: 10.18653/v1/P19-3007.
- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O. and Bowman, S. (2018), "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding", in Linzen, T., Chrupała, G. and Alishahi, A. (Eds.), *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, presented at the EMNLP 2018, Association for Computational Linguistics, Brussels, Belgium, pp. 353–355, doi: 10.18653/v1/W18-5446.
- Wang, B. and Kuo, C.-C.J. (2020), "SBERT-WK: A Sentence Embedding Method by Dissecting BERT-Based Word Models", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, presented at the IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 28, pp. 2146–2157, doi: 10.1109/TASLP.2020.3008390.
- Wang, C. and Banko, M. (2021), "Practical Transformer-based Multilingual Text Classification", in Kim, Y., Li, Y. and Rambow, O. (Eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, presented at the NAACL-HLT 2021, Association for Computational Linguistics, Online, pp. 121–129, doi: 10.18653/v1/2021.naacl-industry.16.
- Wang, Q., Ma, Y., Zhao, K. and Tian, Y. (2022), "A Comprehensive Survey of Loss Functions in Machine Learning", *Annals of Data Science*, Vol. 9 No. 2, pp. 187–212, doi: 10.1007/s40745-020-00253-5.
- Weerts, H.J.P., Mueller, A.C. and Vanschoren, J. (2020), "Importance of tuning hyperparameters of machine learning algorithms", *CoRR*, Vol. abs/2007.07588.

- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., et al. (2023), “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”, arXiv, 10 January, doi: 10.48550/arXiv.2201.11903.
- Weng, L. (2021), “Contrastive representation learning”, Lilianweng.Github.io.
- Wieting, J., Bansal, M., Gimpel, K. and Livescu, K. (2016), “Towards Universal Paraphrastic Sentence Embeddings”, arXiv, 4 March, doi: 10.48550/arXiv.1511.08198.
- Wilbur, W.J. and Sirotkin, K. (1992), “The Automatic Identification of Stop Words”, *Journal of Information Science*, SAGE Publications Ltd, Vol. 18 No. 1, pp. 45–55, doi: 10.1177/016555159201800106.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., et al. (2020), “Transformers: State-of-the-Art Natural Language Processing”, in Liu, Q. and Schlangen, D. (Eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, Online, pp. 38–45, doi: 10.18653/v1/2020.emnlp-demos.6.
- Working Group on Commercial Fraud. (2010), *Commercial Fraud Typologies Summary*, pp. 1–30.
- World Customs Organisation. (2017), *WCO Handbook on Inward and Outward Processing Procedures*.
- World Customs Organization. (2012), “HS Nomenclature 2012 Edition”.
- World Customs Organization. (2017), *WCO Study Report on Cross-Border E-Commerce*.
- World Customs Organization. (2018a), *WCO Annual Report 2017-2018*.
- World Customs Organization. (2018b), *Harmonized System Compendium – 30 Years On*.
- World Customs Organization. (2018c), “Glossary of international customs terms”.
- World Customs Organization. (2022), “General Rules for the Interpretation of the Harmonized System”, 28 January, available at: https://www.wcoomd.org/-/media/wco/public/global/pdf/topics/nomenclature/instruments-and-tools/hs-interpretation-general-rules/0001_2012e_gir.pdf?la=en (accessed 19 June 2025).
- World Customs Organization. (2024a), *WCO Annual Report 2023-2024*.

- World Customs Organization. (2024b), *Exploratory Study on a Possible Strategic Review of the HS*.
- World Trade Organization. (2024), *Global Trade Outlook and Statistics*.
- Wu, Z., Wang, S., Gu, J., Khabsa, M., Sun, F. and Ma, H. (2020), "CLEAR: Contrastive Learning for Sentence Representation", *arXiv*, 31 December.
- Xiang, R., Chersoni, E., Long, Y., Lu, Q. and Huang, C.-R. (2020), "Lexical Data Augmentation for Text Classification in Deep Learning", in Goutte, C. and Zhu, X. (Eds.), *Advances in Artificial Intelligence*, Springer International Publishing, Cham, pp. 521–527, doi: 10.1007/978-3-030-47358-7_53.
- Xiao, C., Long, Y. and Al Moubayed, N. (2023), "On Isotropy, Contextualization and Learning Dynamics of Contrastive-based Sentence Representation Learning", in Rogers, A., Boyd-Graber, J. and Okazaki, N. (Eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, presented at the Findings 2023, Association for Computational Linguistics, Toronto, Canada, pp. 12266–12283, doi: 10.18653/v1/2023.findings-acl.778.
- Xu, H., Pang, G., Wang, Y. and Wang, Y. (2023), "Deep Isolation Forest for Anomaly Detection", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35 No. 12, pp. 12591–12604, doi: 10.1109/TKDE.2023.3270293.
- Xu, L., Xie, H., Li, Z., Wang, F.L., Wang, W. and Li, Q. (2023), "Contrastive Learning Models for Sentence Representations", *ACM Transactions on Intelligent Systems and Technology*, Vol. 14 No. 4, p. 67:1-67:34, doi: 10.1145/3593590.
- Xu, Y., Gábor, K., Milleret, J. and Segond, F. (2023), "Comparative Analysis of Anomaly Detection Algorithms in Text Data", in Mitkov, R. and Angelova, G. (Eds.), *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, presented at the RANLP 2023, INCOMA Ltd., Shoumen, Bulgaria, Varna, Bulgaria, pp. 1234–1245.
- Yang, K., Kim, W.-J., Yang, J. and Kim, Y. (2013), "Ontology matching for recommendation of HS code".

- Yang, K., Kpotufe, S. and Feamster, N. (2021), "An Efficient One-Class SVM for Anomaly Detection in the Internet of Things", *arXiv*, 22 April, doi: 10.48550/arXiv.2104.11146.
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. and Le, Q.V. (2019), "XLNet: generalized autoregressive pretraining for language understanding", *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Curran Associates Inc., Red Hook, NY, USA, pp. 5753–5763.
- Yokkampon, U., Chumkamon, S., Mowshowitz, A., Fujisawa, R. and Hayashi, E. (2020), "Improved Variational Autoencoder Anomaly Detection in Time Series Data", *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE Press, Toronto, ON, pp. 82–87, doi: 10.1109/SMC42975.2020.9283010.
- Yoon Kim. (2014), "Convolutional Neural Networks for Sentence Classification", in Moschitti, A., Pang, B. and Daelemans, W. (Eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, presented at the EMNLP 2014, Association for Computational Linguistics, Doha, Qatar, pp. 1746–1751, doi: 10.3115/v1/D14-1181.
- Zangari, A., Marcuzzo, M., Rizzo, M., Giudice, L., Albarelli, A. and Gasparetto, A. (2024), "Hierarchical Text Classification and Its Foundations: A Review of Current Research", *Electronics*, Multidisciplinary Digital Publishing Institute, Vol. 13 No. 7, p. 1199, doi: 10.3390/electronics13071199.
- Zaraini, M.N. and Yusop, N. (2025), "How Many Keywords are Enough? Determining the Optimal Top-K for Educational Website Classification", *International Journal of Research and Innovation in Social Science*, RSIS International, Vol. IX No. VI, pp. 1574–1586, doi: 10.47772/ijriss.2025.906000126.
- Zareapoor, M., Shamsolmoali, P., Kumar Jain, D., Wang, H. and Yang, J. (2018), "Kernelized support vector machine with deep learning: An efficient approach for extreme multiclass dataset", *Pattern Recognition Letters*, Vol. 115, pp. 4–13, doi: <https://doi.org/10.1016/j.patrec.2017.09.018>.
- Zaub. (2018), "India Imports 2016-2018", 1 January.

- Zhang, D., Li, S.-W., Xiao, W., Zhu, H., Nallapati, R., Arnold, A.O. and Xiang, B. (2021), *“Pairwise Supervised Contrastive Learning of Sentence Representations”*, in Moens, M.-F., Huang, X., Specia, L. and Yih, S.W. (Eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, presented at the EMNLP 2021, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, pp. 5786–5798, doi: 10.18653/v1/2021.emnlp-main.467.
- Zhang, J., Zhang, L., Zhao, Z., Liu, Y., Gu, J., Li, Q. and Zhang, D. (2014), *“Joint Shape and Texture Based X-Ray Cargo Image Classification”*, *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, IEEE Computer Society, USA, pp. 266–273, doi: 10.1109/CVPRW.2014.48.
- Zhang, M., Xu, B. and Gong, J. (2015), *“An Anomaly Detection Model Based on One-Class SVM to Detect Network Intrusions”*, 2015 11th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN), presented at the 2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN), IEEE, Shenzhen, China, pp. 102–107, doi: 10.1109/MSN.2015.40.
- Zhang, P., Zheng, L., Yan, R., Li, C., Hu, R., Zhou, S. and Liu, Q. (2022), *“Multiclass product classification based on multilingual model and LightGBM (team: Uni)”*.
- Zhang, W., Lu, Y., Dubrov, B., Xu, Z., Shang, S. and Maldonado, E. (2021), *“Deep Hierarchical Product Classification Based on Pre-Trained Multilingual Knowledge”*, *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*.
- Zhang, X., Zhao, J. and LeCun, Y. (2016), *“Character-level Convolutional Networks for Text Classification”*, *arXiv*, 4 April, doi: 10.48550/arXiv.1509.01626.
- Zhang, Y., Jin, R. and Zhou, Z.-H. (2010), *“Understanding Bag-of-Words Model: A Statistical Framework”*, *International Journal of Machine Learning and Cybernetics*, Vol. 1 No. 1, pp. 43–52, doi: 10.1007/s13042-010-0001-0.
- Zhang, Y. and Wallace, B. (2017), *“A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification”*, in Kondrak, G. and Watanabe, T. (Eds.), *Proceedings of the Eighth International Joint Conference on*

- Natural Language Processing (Volume 1: Long Papers)*, presented at the IJCNLP 2017, Asian Federation of Natural Language Processing, Taipei, Taiwan, pp. 253–263.
- Zheng, A. and Casari, A. (2018), *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*, O'Reilly Media, Inc.
- Zhou, C., Che, C., Zhang, X.S., Zhang, Q. and Zhou, D. (2022), “Harmonized System Code Prediction of Import and Export Commodities Based on Hybrid Convolutional Neural Network with Auxiliary Network”, *Knowledge-Based Systems*, Elsevier, Amsterdam, Vol. 256, p. 109836, doi: 10.1016/j.knosys.2022.109836.
- Zhou, X., Zhu, Y., Li, Y., Qiang, J., Yuan, Y., Wu, X. and Zhang, R. (2023), “A hybrid classification method via keywords screening and attention mechanisms in extreme short text”, *Intell. Data Anal.*, Vol. 27 No. 5, pp. 1331–1345, doi: 10.3233/IDA-220417.
- Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., et al. (2021), “A Comprehensive Survey on Transfer Learning”, *Proceedings of the IEEE*, presented at the *Proceedings of the IEEE*, Vol. 109 No. 1, pp. 43–76, doi: 10.1109/JPROC.2020.3004555.