University of Zurich[UZH]

Master's Thesis
presented to the Faculty of Business, Economics and Informatics
of the University of Zurich
for the degree of
Master of Science in Informatics

Exploring Text Mining Applications in Financial Negotiations
A Case Study on Synthetic Data

Author: Kevin Bründler
Student ID: 19-919-273

Examiner: Prof. Dr. Martin Volk
Supervisor: Dr. Simon Clematide

Department of Informatics

Submission Date: 11 July 2025

Abstract

Synthetic data presents a promising approach for advancing Natural Language Processing (NLP) in domains where real-world communications are inaccessible due to privacy and commercial sensitivity. Maritime chartering exemplifies this challenge, characterized by specialized terminology, confidential email negotiations, and high-value transactions. This thesis presents a comprehensive pipeline that addresses data constraints through the generation of synthetic data and subsequent information extraction for maritime entities, including vessels, ports, commodities, and Incoterms. We assess different generation methods across five large language models (LLMs) (Claude, GPT4, Gemini, DeepSeek, and Mistral), finding that the two-stage Base-Refine (BARE) method is most effective, producing emails that achieve 85% of human vocabulary diversity and match human trigram diversity. However, this success comes with notable limitations. Current LLMs exhibit a positivity bias, systematically under-representing negative sentiment in generated content. This bias intensifies during BARE's refinement step, which actively suppresses conflict, skewing communications towards a tone that diverges from the more balanced sentiment found in real business emails. Furthermore, authorship detection experiments using DetectGPT revealed an inversion of the standard hypothesis, suggesting that detection models may be confounded by maritime vocabulary and data leakage.

These synthetic datasets reveal a fundamental trade-off: higher-quality synthetic data, measured by linguistic diversity, creates a more challenging extraction task. We demonstrate this by comparing span-based (GLiNER) and generative (Qwen) models on over 19,000 synthetic email chains. While both architectures benefit from finetuning, GLiNER proves more robust, progressing from a zero-shot F1 score of 0.66 to a peak of 0.86 through contrastive finetuning with hard negative sampling. In contrast, the generative model achieves similar zero-shot performance (0.66) but shows greater sensitivity to entity co-occurrence patterns, resulting in higher variability after finetuning. Finally, our experiments challenge existing literature on the superiority of code-based models for information extraction. We demonstrate that small-scale, code-specialized LLMs (fewer than 3B parameters) fail at this task, indicating a capability threshold below which deficits in natural language understanding outweigh their structured reasoning capabilities. Ultimately, we demonstrate that realism in synthetic data is no free lunch, as it carries embedded bias and increased task complexity that require careful navigation.

Our code is accessible at `github.com/random9nessss/master_thesis_ifi`

Zusammenfassung

Synthetische Daten stellen einen vielversprechenden Ansatz dar, um die Verarbeitung natürlicher Sprache (Natural Language Processing, NLP) in Bereichen voranzutreiben, in denen die reale Kommunikation aufgrund von Datenschutz und kommerzieller Sensibilität unzugänglich ist. Das Chartern von Schiffen ist ein Beispiel für diese Herausforderung, die durch spezielle Terminologie, vertrauliche E-Mail-Verhandlungen und hochwertige Transaktionen gekennzeichnet ist. In dieser Arbeit wird eine umfassende Pipeline vorgestellt, die Datenbeschränkungen durch die Generierung synthetischer Daten und die anschliessende Informationsextraktion für maritime Einheiten, einschliesslich Schiffen, Häfen, Waren und Incoterms, angeht. Wir bewerten verschiedene Generierungsmethoden mit fünf grossen Sprachmodellen (Claude, GPT4, Gemini, DeepSeek und Mistral) und stellen fest, dass die zweistufige Base-Refine-Methode (BARE) am effektivsten ist und E-Mails erzeugt, die 85% der menschlichen Wortschatzvielfalt erreichen und der menschlichen Trigrammvielfalt entsprechen. Dieser Erfolg ist jedoch mit signifikanten Einschränkungen verbunden. Aktuelle Sprachmodelle weisen einen Positivitäts-Bias auf, d. h. negative Stimmungen werden in den generierten Inhalten systematisch unterrepräsentiert. Diese Verzerrung verstärkt sich während des Verfeinerungsschritts von BARE, der Konflikte aktiv unterdrückt und die Kommunikation in eine Richtung lenkt, die von der ausgewogenen Stimmung in echten Geschäfts-E-Mails abweicht. Darüber hinaus ergaben Experimente zur Erkennung der Urheberschaft mit DetectGPT eine Umkehrung der Standardhypothese, was darauf hindeutet, dass die Erkennungsmodelle durch maritimes Vokabular und Datenkontamination verwirrt werden können.

Unsere synthetischen Datensätze offenbaren einen grundlegenden Kompromiss: Qualitativ hochwertigere synthetische Daten, gemessen an der linguistischen Vielfalt, stellen eine grössere Herausforderung für die Extraktion dar. Wir demonstrieren dies durch den Vergleich von spannenbasierten (GLiNER) und generativen (Qwen) Modellen anhand von über 19.000 synthetischen E-Mail-Ketten. Während beide Architekturen von der Feinabstimmung profitieren, erweist sich GLiNER als robuster, da es sich durch kontrastives Finetuning mit hartem negativem Sampling von einem F1-Score von 0,66 auf einen Spitzenwert von 0,86 steigern konnte. Im Gegensatz dazu erreicht das generative Modell eine ähnliche Zero-Shot-Performance (0,66), zeigt aber eine grössere Empfindlichkeit gegenüber Entity-Co-Occurrence-Mustern, was zu einer höheren Variabilität nach dem Finetuning führt. Schliesslich stellen unsere Experimente die bestehende Literatur über die Überlegenheit codebasierter Modelle bei der Informationsextraktion in Frage. Wir zeigen, dass kleine, auf Codes spezialisierte LLMs (weniger als 3B-Parameter) bei dieser Aufgabe scheitern, was auf eine Fähigkeitsschwelle hindeutet, unterhalb derer Defizite im Verständnis

natürlicher Sprache ihre Fähigkeiten zum strukturierten Schlussfolgern überwiegen. Letztendlich zeigen wir, dass Realismus in synthetischen Daten nicht umsonst zu haben ist, da er mit Verzerrungen und erhöhter Aufgabenkomplexität einhergeht, die eine sorgfältige Navigation erfordern.

Der Code wird unter `github.com/random9nessss/master_thesis_ifi` veröffentlicht.

Acknowledgment

I want to express my sincere gratitude to all those who supported me throughout the completion of this thesis. First and foremost, I am grateful to my supervisor, Dr. Simon Clematide, for his invaluable mentorship. His dedication to our weekly meetings, insightful guidance, and knowledge sharing have been instrumental in this research. I would also like to thank Prof. Dr. Martin Volk, Professor of Computational Linguistics at the University of Zurich, for providing me with the opportunity to conduct this research. Additionally, I appreciate the support and encouragement of my family and friends throughout this challenging yet rewarding process. Finally, I would also like to thank the University of Zurich for providing an excellent academic environment and an intellectually stimulating curriculum during my master's studies. The knowledge and skills acquired through the program have been fundamental to completing this research.

Disclaimer

This thesis was completed with the assistance of generative AI tools, specifically for paraphrasing, grammar correction, and code documentation. While these tools provided valuable support in refining the presentation of ideas, all scientific content, research findings, analysis, and conclusions presented in this work are entirely my own.

Contents

Abstract ii

Acknowledgment v

Disclaimer vi

List of Figures xii

List of Tables xiv

List of Acronyms xvi

List of Maritime Acronyms xvii

Maritime Glossary xviii

1. Introduction 1

1.1. Motivation 1

1.1.1. Maritime Chartering as a Case Study 2

1.2. Research Questions 3

1.3. Thesis Structure 4

2. Background and Related Work 6

2.1. Background 6

2.1.1. A Primer in Maritime Chartering 6

2.1.1.1. The Chartering Process Workflow 6

2.1.2. Synthetic Data 7

2.1.2.1. Attribute prompting (AttrPrompting) 8

2.1.2.2. Base-Refine (BARE) 9

2.1.3. Text Mining 10

2.1.3.1. Named Entity Recognition (NER) 10

2.1.3.2. Generative Information Extraction 11

2.1.3.3. Template Filling 13

2.2. Related Work 14

2.2.1. Synthetic Email Data 14

2.2.2. Text Mining on Emails 15
2.2.3. Artificial Intelligence in Maritime Chartering 16

3. Datasets **17**
3.1. Email Chains 18
3.1.1. Synthetic Generations 18
3.1.2. Enron Corpus 20
3.2. Datasets for Email Generation 20
3.2.1. Port Information 21
3.2.2. Vessel Information 21
3.2.3. Names Dataset 22
3.2.4. Maritime Chartering Datasets 22
3.2.5. News Articles 22
3.2.6. Wikipedia Data Collection 23
3.2.7. International Maritime Organization Vessel Codes 24
3.2.8. UN/LOCODE Code List by Country and Territory 25
3.3. Natural Language Inference for SimCSE 25
3.4. NER Datasets 26
3.4.1. PileNER 26
3.4.2. Template-Generated Maritime Training Data 27
3.4.3. Out-of-Distribution (OOD) NER Benchmark 28
3.5. Massive Text Embedding Benchmark (MTEB) 28

4. Methodology **30**
4.1. Synthetic Data 30
4.1.1. Data Generation Methods 30
4.1.1.1. Attribute Prompting (AttrPrompting) 30
4.1.1.2. Base-Refine (BARE) 31
4.1.1.3. Zero-Shot Prompting 33
4.1.1.4. Few-Shot Prompting 33
4.1.2. Evaluation and Analysis 33
4.1.2.1. Textual Diversity 34
4.1.2.2. Sentiment 35
4.1.2.3. Verbosity 36
4.1.3. Authorship Detection 36
4.2. Text Mining 37
4.2.1. Evaluation Metric: Macro F1 38
4.2.1.1. Precision, Recall, and F1 Score 38
4.2.2. TF-IDF Extraction 39

4.2.3. GLiNER-based NER Models 40
4.2.3.1. Stage 1: Encoder Pretraining Strategies 41
4.2.3.2. Stage 2: GLiNER Training and Finetuning 46
4.2.3.3. Summary of Model Variants 48
4.2.4. Generative Information Extraction 49
4.2.4.1. Code-Style vs. Standard Prompting Approaches . . . 49
4.2.4.2. Template Filling: Qwen-0.5B 51
4.3. Embedding Analysis 52

5. Results and Discussion **54**
5.1. Synthetic Data 54
5.1.1. Enhanced Diversity Through BARE 54
5.1.2. Comparative Analysis with a Human Reference Corpus 57
5.1.3. Sentiment Distribution Analysis 58
5.1.3.1. Impact of Refinement on Sentiment 59
5.1.3.2. GPT4's Sentiment Transformation 60
5.1.4. Verbosity Patterns in Generated Emails 64
5.1.5. Refine-as-you-go or Refine-at-once? 65
5.1.6. Authorship Detection 68
5.1.7. Practical Implications for Deployment 69
5.1.7.1. Generation Method Performance Profiles 69
5.1.7.2. Model-Level Performance Analysis 71
5.2. Text Mining 72
5.2.1. GLiNER: Span-based Entity Recognition 73
5.2.1.1. Encoder Pretraining Analysis 73
5.2.1.2. Out of Domain Performance 77
5.2.1.3. Maritime Domain Performance 80
5.2.2. CodeIE vs. Standard LLM Performance Analysis 84
5.2.3. Qwen-2.5-0.5B: Template Filling 86
5.2.3.1. The BARE Challenge: Entity Co-Occurrence 88
5.2.4. Comparative Analysis of GLiNER and Qwen 90

6. Limitations and Future Work **93**
6.1. Limitations 93
6.2. Future Work 94

7. Conclusion **96**
7.1. Revisiting the Research Questions 96
7.2. Synthetic Data Generation: Methods, Models and Trade-offs 96
7.3. The Diversity-Extractability Relationship 98

7.4. Closing Remarks 100

References **100**

A. Synthetic Data Generation **112**

A.1. Attribute Prompting (AttrPrompting) 112
A.1.1. Email Attributes 112
A.1.1.1. Vessel Names and Details 112
A.1.1.2. Port Names 113
A.1.1.3. Names 113
A.1.1.4. Incoterms 113
A.1.1.5. Commodities 114
A.1.1.6. Chartering Abbreviations 115
A.1.1.7. Charterparty Terms 115
A.1.1.8. Brokers 116
A.1.1.9. Formality Level 116
A.1.1.10. Level of English 117
A.1.1.11. Verbosity Levels 117
A.1.1.12. Type of Noise 118
A.1.1.13. Conversation Mode 118
A.1.1.14. Anchor Date 118
A.1.1.15. Cargo Size 118
A.1.1.16. Broker Demeanor 119
A.1.1.17. Email Count 119
A.1.2. Prompt 119
A.2. Base-Refine (BARE) 122
A.2.1. Sample Seed Emails for BARE Generation 122
A.2.2. Base Generation 123
A.2.2.1. Starting Prompt (Few-Shot) 124
A.2.2.2. Iterative Prompt 124
A.2.3. Refinement 126
A.2.3.1. Refinement Prompt 126
A.3. Zero-Shot Prompting 128
A.3.1. Prompt 128
A.4. Few-Shot Prompting 130
A.4.1. Prompt 130
A.5. API Interaction Costs 131
A.6. Maritime Entity Annotation Prompt 132

B. DetectGPT **134**

C. Text Embeddings **136**

D. OOD Evaluation **139**

E. GLiNER **141**

E.1. Entity-Specific Performance Analysis 141

E.2. Performance Distribution Analysis 141

F. CodeIE vs. Standard LLM Performance Analysis **146**

G. Qwen-2.5-0.5B: Template Filling **147**

G.1. Entity-Specific Performance Heatmaps 147

G.1.1. Entity-Specific Performance by Generation Model 147

G.1.2. Overall Performance Distributions 148

G.2. Detailed BARE Generation Results 148

List of Figures

1.1. Overview of Vessels in Transit 3

2.1. Maritime Chartering Process Flow 7
2.2. Architectural Overview of the GLiNER Model 12

3.1. Word Cloud of Collected News Articles 23
3.2. Word Cloud of Collected Wikipedia Articles 24
3.3. Map of Scraped Ports from UN/LOCODE 25
3.4. PileNER Label Consolidation 27

4.1. GLiNER Training Pipeline 41

5.1. Word Cloud Comparison of Vocabulary Distribution 57
5.2. Sentiment Distribution of Synthetic Data 59
5.3. Change in Negative Sentiment Through Refinement 60
5.4. Change in Positive Sentiment Through Refinement 61
5.5. Sentiment Transformation Flow for GPT4 Refinement 61
5.6. DetectGPT Authorship Detection 70
5.7. t-SNE Visualization of Entity Embeddings (Politics) 75
5.8. t-SNE Visualization of Entity Embeddings (AI) 76
5.9. t-SNE Visualization of Entity Embeddings (Ports and Vessels) 76
5.10. Heatmap Overall F1 Scores for Maritime GLiNER Evaluation 81
5.11. GLiNER Model Performance by Finetuning Strategy 81
5.12. Breakdown of GLiNER Performance by Generation Factors 83
5.13. Port vs. Vessel F1 Scores for Zero-Shot Models 83
5.14. GLiNER F1 Scores for Incoterm Entity Extraction 84
5.15. Performance Comparison of Code and Standard LLMs 85
5.16. Extraction Heatmap Qwen Overall 86
5.17. Entity Extraction Performance Distribution Qwen Base 87
5.18. Extraction Heatmap Qwen Incoterm 88
5.19. Extraction Heatmap Qwen Vessel 89
5.20. Qwen and GLiNER Comparison 91

B.1. DetectGPT Authorship Detection 135

C.1. t-SNE Visualization of Entity Embeddings (Literature) 136
C.2. t-SNE Visualization of Entity Embeddings (Movie) 137
C.3. t-SNE visualization of entity embeddings for the Music Dataset . . . 137
C.4. t-SNE Visualization of Entity Embeddings (Restaurant) 138
C.5. t-SNE Visualization of Entity Embeddings (Science) 138

E.1. GLiNER Performance on Vessel Entity Extraction 142
E.2. GLiNER Performance on Port Entity Extraction 143
E.3. GLiNER Performance on Commodity Entity Extraction 144
E.4. GLiNER Performance Distribution by Entity Type 145

G.1. Extraction Heatmap Qwen Commodity 147
G.2. Extraction Heatmap Qwen Port 148
G.3. Entity Extraction Performance Qwen 149
G.4. Overall Extraction Performance Qwen 151

List of Tables

3.1. Overview of Datasets Used in this Thesis 17
3.2. Synthetic Email Chain Generation Counts 18
3.3. Maritime News Sources and Article Counts 22
3.4. Sample Vessel Names from Cleaned IMO Dataset 25
3.5. Examples of NER Label Consolidations 27
3.6. Statistics of the OOD NER Benchmark 28

4.1. TF-IDF Vectorizer Configuration for Domain Term Identification . . 39
4.2. Entity-Specific Indicators and Extraction Patterns 40
4.3. Similarity Strata for Stratified Batch Sampling 42
4.4. Batch Composition for Stratified Contrastive Sampling 43
4.5. Entity-Contrastive Training Hyperparameters 44
4.6. SimCSE Training Hyperparameters 45
4.7. MLM Training Hyperparameters 46
4.8. GLiNER Training Hyperparameters 47
4.9. GLiNER Finetuning Hyperparameters Summary 47
4.10. Overview of GLiNER Model Variants 49
4.11. LoRA Finetuning Configuration for Qwen-0.5B 51
4.12. Training Hyperparameters for Qwen-0.5B Finetuning 52

5.1. Textual Diversity Metrics Across Generation Methods and Models . . 56
5.2. Verbosity Metrics Across Generation Methods 65
5.3. Textual Diversity Iterative BARE vs. Standard BARE 66
5.4. Sentiment Distribution Iterative BARE vs. Standard BARE 67
5.5. Verbosity Metrics Iterative BARE vs. Standard BARE 67
5.6. Clustering Evaluation Results (Seven NER Domains) 73
5.7. MTEB Evaluation Results (Five Pretraining Strategies) 77
5.8. Overall F1 Scores by GLiNER Variant OOD 78
5.9. Entity-level F1 Scores (Restaurant) 78
5.10. Entity-level F1 Scores (Music) 79
5.11. Entity-level F1 Scores (AI) 80
5.12. Precision-Recall Analysis BARE Generations 90

A.1. API Interaction Costs for Synthetic Email Generation 132

D.1. Entity-level F1 Scores (Literature) 139
D.2. Entity-level F1 Scores (Movie) 140
D.3. Entity-level F1 Scores (Politics) 140
D.4. Entity-level F1 Scores (Science) 140

F.1. Performance Comparison Code and Standard LLMs on Maritime Entity Extraction 146

G.1. Qwen Model Performance on BARE 150

List of Acronyms

AttrPrompting	Attribute Prompting
BARE	Base-Refine
CH	Calinski-Harabasz
CRFs	Conditional Random Fields
IE	Information Extraction
LDA	Latent Dirichlet Allocation
LLMs	Large Language Models
LoRA	Low-Rank Adaptation
MLM	Masked Language Model
MTEB	Massive Text Embedding Benchmark
NER	Named Entity Recognition
NLI	Natural Language Inference
NLP	Natural Language Processing
OOD	Out-of-Distribution
PEFT	Parameter-Efficient Finetuning
PLMs	Pretrained Language Models
SimCSE	Simple Contrastive Learning of Sentence Embeddings
STS	Semantic Textual Similarity
t-SNE	t-distributed Stochastic Neighbor Embedding
TF-IDF	Term Frequency-Inverse Document Frequency
TTR	Type-Token Ratio

List of Maritime Acronyms

CFR	Cost and Freight
CIF	Cost, Insurance, and Freight
CP	Charter Party
DAP	Delivered at Place
DDP	Delivered Duty Paid
DWT	Deadweight Tonnage
FOB	Free on board
GAFTA	Grain and Feed Trade Association
IMO	International Maritime Organization
Incoterms	International Commercial Terms
LNG	Liquefied Natural Gas
LOA	Length Overall
LPG	Liquefied Petroleum Gas
SHINC	Sundays and Holidays Included

Maritime Glossary

Note: The definitions in this glossary have been adapted from Wikipedia and other maritime reference sources.

Incoterms (International Commercial Terms)

Incoterms Incoterms constitute a set of international commercial terms published by the International Chamber of Commerce. They define the responsibilities and risks of sellers and buyers for the delivery of goods. The latest version, Incoterms 2020, includes 11 terms.

CFR (Cost and Freight) Under CFR, the seller arranges and pays for the transportation of the goods to the port of destination. However, the seller's risk ends once the goods are loaded on board the vessel. The buyer assumes responsibility for loss or damage once the cargo has been loaded onto the ship.

CIF (Cost, Insurance and Freight) Similar to CFR, except that under CIF terms, the seller also arranges for maritime insurance. The seller's responsibility extends until the goods are safely onboard, while any damage or loss after that point falls to the buyer (although insurance coverage continues through transport).

DAP (Delivered at Place) The seller is responsible for delivering the goods to a specified location in the destination country, bearing all costs and risks of transportation (excluding import duties and local taxes). Once the goods are at the named place, the buyer assumes any further costs and risks.

DDP (Delivered Duty Paid) Under DDP, the seller assumes the maximum obligation, covering all costs and risks of delivering the goods to the buyer's specified destination, including customs duties and import taxes. The buyer's responsibility effectively begins once the items have been delivered and made available for unloading.

FOB (Free on Board) With FOB, the seller takes on the cost and risk of delivering goods aboard the vessel at the port of shipment. Once the goods are "on

board," the risk shifts to the buyer. All subsequent transportation and insurance costs typically fall to the buyer, although variations may exist in specific contracts.

Vessel Specifications and Types

DWT (Deadweight Tonnage) The maximum weight a vessel can safely carry, including cargo, fuel, crew, and passengers. It represents the difference between the vessel's displacement when fully loaded and when empty, serving as a key indicator of a ship's cargo-carrying capacity.

LOA (Length Overall) The maximum length of a vessel. This measurement is crucial for determining berth requirements and canal transit fees.

M/V (Motor Vessel) A designation prefix indicating that a ship is powered by diesel or combustion engines, as opposed to steam turbines (SS) or other propulsion methods, and commonly used in vessel names, e.g., *MV Ocean Star*.

M/T (Motor Tanker) A vessel designed explicitly for transporting liquid cargo, powered by motor engines. These ships feature specialized tank arrangements and pumping systems for loading and discharging liquid commodities.

Bulk Carrier A merchant vessel designed to transport unpackaged cargo such as grains, coal, ore, and cement. These vessels feature large cargo holds with wide hatch openings, facilitating loading and unloading operations.

Tanker A vessel designed for the transport of liquids or liquefied gases. Types include crude oil tankers, chemical tankers, and gas carriers for liquefied natural gas (LNG) or liquefied petroleum gas (LPG).

Cargo and Commodities

LNG (Liquefied Natural Gas) Natural gas is cooled to approximately -162°C to convert it into a liquid form, making it easier to store and transport. LNG carriers require specialized cryogenic tanks and handling systems.

LPG (Liquefied Petroleum Gas) A mixture of propane and butane gases liquefied under pressure for transportation. LPG carriers feature pressurized or refrigerated tanks designed to maintain the cargo in a liquid state during the voyage.

MT (Metric Ton) A unit of mass equal to 1,000 kilograms. The standard unit for measuring cargo weight in maritime shipping is used in freight calculations and vessel capacity specifications.

Commodity Any raw material or primary agricultural product that can be bought and sold in bulk. In a maritime context, common commodities include crude oil, grains, coal, iron ore, and various chemicals, which are transported by specialized vessels.

Charter Party and Commercial Terms

CP (Charter Party) A contract between a shipowner and a charterer for the hire of a vessel or space aboard a vessel. Types include voyage charter (for a specific journey), time charter (for a specific period), and bareboat charter (vessel lease without crew).

SHINC (Sundays and Holidays Included) A charter party clause indicating that Sundays and holidays count as laytime for loading or discharging operations.

Freight Rate The price charged for transporting cargo is typically expressed per metric ton or as a lump sum. Rates vary based on cargo type, voyage distance, vessel size, market conditions, and specific terms negotiated in the charter party.

Demurrage Compensation paid by the charterer to the shipowner when the vessel is detained beyond the agreed laytime for loading or discharging and calculated as a daily rate specified in the charter party.

Despatch A rebate paid by the shipowner to the charterer when cargo operations are completed before the expiry of the agreed laytime. Typically calculated at half the demurrage rate, this encourages efficient cargo handling and rewards charterers for a quick turnaround.

Laytime The amount of time agreed between the shipowner and the charterer for loading and/or discharging cargo. Exceeding laytime results in demurrage charges, while completing operations early may earn despatch.

Organizations and Standards

IMO (International Maritime Organization) A specialized agency of the United Nations responsible for regulating shipping. The IMO develops and maintains

a comprehensive regulatory framework for shipping, covering safety, environmental concerns, legal matters, and maritime security.

GAFTA (Grain and Feed Trade Association) An international trade association that provides standard contract forms for the grain and feed industries. GAFTA contracts are widely used in the maritime shipping of agricultural commodities, including grains and soybeans.

1. Introduction

While algorithms and models capture headlines, the unseen engine of modern commerce remains the humble email. It is the primary vehicle for negotiation, contracting, and logistics, yet its contents remain a black box for data-driven analysis due to privacy and commercial sensitivity. This thesis aims to illuminate that black box by creating an end-to-end pipeline: synthetic data generation methods that produce authentic maritime email correspondence, coupled with text mining techniques that can extract critical business information.

1.1. Motivation

While Large Language Models (LLMs) have demonstrated remarkable capabilities across various sectors, their application within finance remains comparatively underexplored (Lee et al., 2025). Recent proprietary developments, such as BloombergGPT (Wu et al., 2023), highlight the recognition of LLMs' potential value in financial applications. However, access to these models is often prohibitively expensive (Liu et al., 2023), limiting their utility to well-resourced organizations. Although emerging open-source initiatives aim to democratize access to finance-oriented LLMs (Yang et al., 2023a,b; Liu et al., 2021a), the lack of high-quality textual data remains a significant challenge. Data accessibility throughout finance is limited due to its sensitive nature, with confidentiality and regulatory constraints often restricting access even for internal stakeholders (Assefa et al., 2020). This lack of data poses a challenge not only for model development but also makes it difficult to assess model performance in real-world settings (Du et al., 2025).

Data scarcity is especially pronounced for email data, which is essential to corporate communication and negotiations. Recent estimates gathered by *Statista* indicate that over 300 billion emails are exchanged worldwide each day,[1], but public access to these data is severely limited due to privacy considerations and commercial interest. While datasets such as the ENRON corpus (Klimt and Yang, 2004)

[1] https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/

exist, they are increasingly outdated and may not capture the complexity of contemporary business communication. As a result, companies would need to manually extract information from their internal communications to build datasets for training. However, manually parsing email data is time-consuming, error-prone, and diverts valuable employee resources from more productive tasks. Advanced text mining methodologies and LLMs can offer a promising, autonomous, and scalable alternative.

1.1.1. Maritime Chartering as a Case Study

Among the various financial subdomains affected by data limitations, maritime chartering presents a particularly compelling case for investigation due to its heavy reliance on email correspondence for freight negotiations, port clearances, and contractual arrangements. This domain features distinct linguistic characteristics, including specialized jargon, technical abbreviations, port codes, and vessel specifications, which pose challenges to generic language processing approaches. At the same time, the significance of the sector cannot be overstated. According to the United Nations Conference on Trade and Development, global maritime trade reached 12.3 billion tons in 2023, growing at an annual rate of 2.4%, and is projected to maintain this growth rate through 2029 (United Nations Conference on Trade and Development, 2024). The number of vessels at sea at any moment underscores the scale of global shipping, as shown in Figure 1.1 with real-time data from VesselFinder (VesselFinder, 2025).

The maritime sector accounts for approximately 80% of global trade by volume, with shipping contributing 3% of global greenhouse gas emissions. The financial implications of chartering decisions have been amplified by recent volatility in freight rates, which surged dramatically in 2024 due to rerouting, port congestion, and rising operational costs (United Nations Conference on Trade and Development, 2024). Despite its central economic role, there exists no publicly available dataset of contractual communications in shipping, a gap reflecting both the commercially sensitive nature of these exchanges and the need for domain-specific resources.

Generic language models may struggle with specialized terminology and contextual nuances of maritime chartering, making it an ideal test case for exploring synthetic data generation and text mining approaches. The development of synthetic yet realistic email datasets can address critical knowledge gaps with significant economic implications. This application is especially relevant as the industry navigates what has been characterized as an "exceptionally daunting operating landscape" shaped

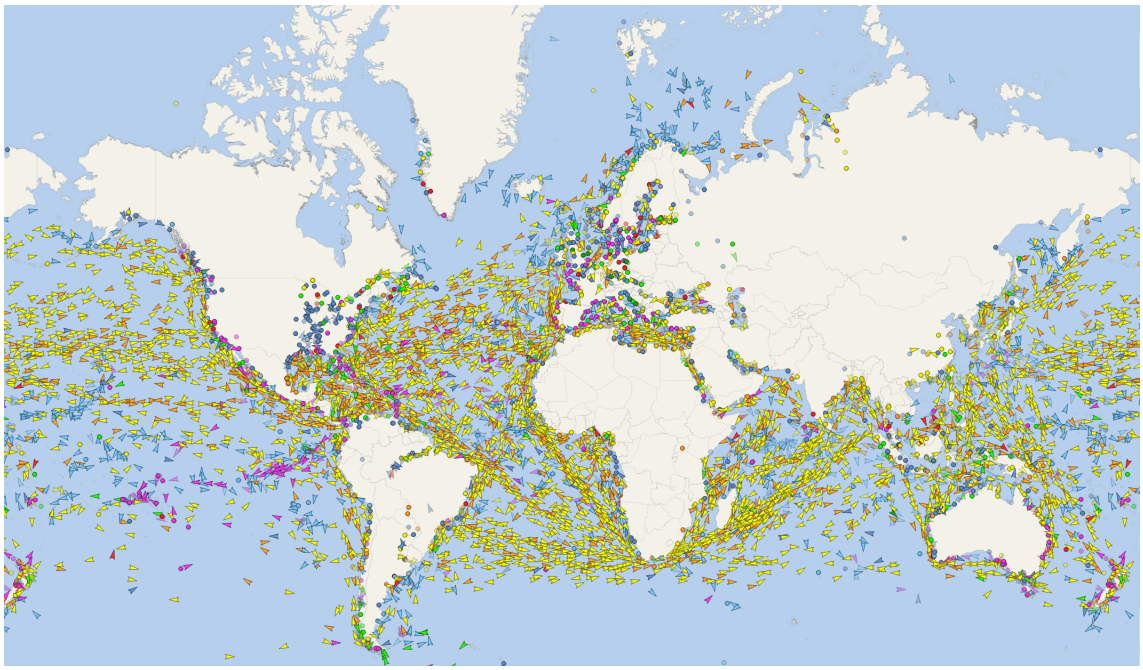

Figure 1.1.: Overview of vessels in transit as of March 10, 2025.

by geopolitical tensions and climate-related disruptions (Potluru et al., 2023).

1.2. Research Questions

The absence of publicly available email corpora for maritime chartering presents both a challenge and an opportunity. In this thesis, we investigate whether synthetic data generation can address this gap while maintaining sufficient quality for downstream text mining applications. Our research is guided by two interconnected themes that lead to the following two research questions:

How can we generate high-quality synthetic maritime emails that capture the linguistic characteristics of authentic business correspondence? This research question addresses the methodologies for generating high-quality synthetic maritime emails that emulate the linguistic characteristics of genuine business correspondence. We first investigate the influence of different prompting strategies and generation methods on the diversity, sentiment, verbosity, and authenticity of their output. Subsequently, a comparative analysis of prominent LLMs (Mistral (Jiang et al., 2024), Deepseek (Liu et al., 2024), Claude (Anthropic, 2024), GPT4 (Achiam et al., 2023), and Gemini (Team et al., 2023)) determines their effectiveness in producing realistic maritime correspondence. Finally, we assess whether multi-stage generation frameworks, specifically base-refine (BARE) introduced by Zhu et al. (2025), offer

an improvement in linguistic characteristics compared to conventional single-stage generation methods.

How does the quality and diversity of synthetic training data impact the performance of information extraction systems? Once the emails have been generated, we investigate the relationship between the characteristics of synthetic data and the effectiveness of downstream information extraction. First, we compare different extraction methodologies to evaluate their relative strengths and limitations when trained on a synthetic corpus. A central component of this thesis is to determine whether higher textual diversity in the synthetic emails leads to increased complexity for the extraction task. Lastly, we investigate the efficacy of various pretraining strategies (contrastive learning, domain adaptation) and finetuning strategies in enhancing model performance on both entity extraction accuracy and disambiguating semantically similar entities.

These research questions are designed not only to address the immediate challenge of data scarcity in shipping but also to provide insights into the broader viability of synthetic data for specialized domain applications. By systematically evaluating both generation quality and extraction performance, we aim to establish best practices for synthetic data creation in domains where real data is scarce.

1.3. Thesis Structure

The following section provides a brief overview of the subsequent chapters and their content.

Chapter 2: Background and Related Work This chapter establishes the theoretical foundation for understanding both the maritime chartering domain and the computational approaches employed in this thesis. It starts with an introduction to maritime chartering processes, providing context for readers unfamiliar with the shipping industry. The chapter then examines state-of-the-art techniques in synthetic data generation, including attribute prompting and base-refine methodologies, before exploring text mining with particular emphasis on NER and generative information extraction. Finally, we survey existing literature at the intersection of artificial intelligence and maritime operations.

Chapter 3: Datasets At the core of this work are 19,817 synthetically generated email chains representing maritime chartering negotiations, created using multiple prompting strategies and language models. Authentic email sources from the Enron corpus complement these artificial communications. The chapter also details supplementary datasets that enhance the realism of generation, including port locations, vessel specifications, and personal names, alongside specialized maritime resources such as news articles and Wikipedia content.

Chapter 4: Methodology This chapter details the technical approaches developed for generating and analyzing synthetic maritime chartering emails. It begins by describing the distinct generation strategies designed to produce realistic email chains with varying degrees of control and diversity. The chapter then outlines the evaluation framework, encompassing metrics for textual diversity, sentiment analysis, verbosity measurement, and authorship detection. Finally, it presents text mining methodologies, including contrastive finetuning approaches for entity disambiguation and various pretraining strategies.

Chapter 5: Results and Discussion Within this chapter, we evaluate synthetic data generation methods and text mining approaches across two main sections. We first examine the quality of generated emails through diversity metrics, sentiment analysis, and authorship detection. The second part analyzes maritime entity extraction using both GLiNER's span-based approach and generative methods, demonstrating the critical importance of domain-specific finetuning and the relationship between textual diversity and extraction difficulty.

Chapter 6: Limitations and Future Work Methodological constraints and future research opportunities are examined across three key dimensions. The limitations section addresses data generation constraints, including hardware restrictions and reliance on the (out)dated Enron corpus as a human baseline, experimental design limitations coming from computational constraints on hyperparameter optimization, and potential label biases. Finally, future research directions are suggested.

Chapter 7: Conclusion This concluding chapter synthesizes the key findings from both synthetic data generation and text mining experiments, demonstrating how different generation methods affect textual diversity and extraction performance in maritime communications. The chapter reviews the core contributions in line with the original research questions and discusses broader implications.

2. Background and Related Work

2.1. Background

To provide the necessary context for this thesis, this chapter offers a high-level introduction to maritime chartering, followed by an examination of the theoretical concepts that underpin the two primary components: data generation methodologies and text mining models.

2.1.1. A Primer in Maritime Chartering

Maritime chartering represents the commercial process where shipowners lease their vessels to charterers for the transportation of goods. The chartering market operates as a marketplace connecting three primary stakeholders: shipowners, charterers, and shipbrokers. Shipowners control vessel assets and seek to maximize revenue through optimal vessel employment. Charterers represent cargo owners, trading companies, or logistics providers requiring transportation capacity for their goods. Shipbrokers act as intermediaries, connecting owners and charterers while providing market intelligence and negotiation services in exchange for commission.

2.1.1.1. The Chartering Process Workflow

The chartering process begins when cargo owners identify transportation requirements. They typically engage shipbrokers with specifications that include cargo type, quantity, loading and discharge ports, and timing constraints. Brokers circulate these requirements to vessel owners, who respond with availability windows and indicative freight offers that brokers relay back to charterers. The negotiation phase proceeds through iterative offer-counteroffer cycles conducted via email, addressing freight rates, laytime (the allowed loading and discharge duration), demurrage, and despatch (monetary penalties or bonuses for time overruns or savings), as well as contract-specific clauses. These negotiations can span days or even weeks, gener-

ating multiple email threads. A fixture occurs once both parties agree to binding terms, typically recapped by email, creating a legal obligation between the charterer and the shipowner. However, several inquiries fail to reach fixture as charterers may submit speculative inquiries to gauge market conditions, vessel availability may not align with cargo timing, vessel dimensions might be unsuitable for entering ports, or negotiations stall over commercial terms.

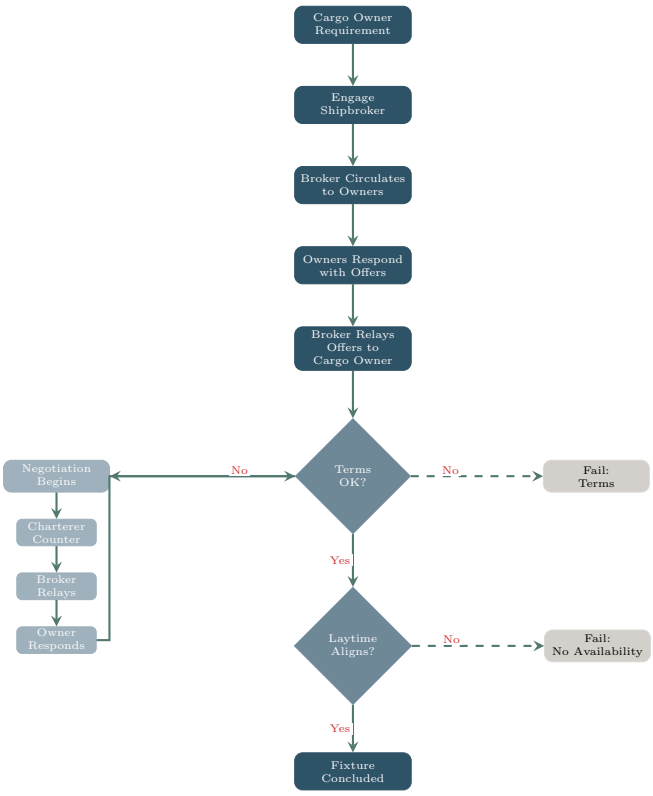

Figure 2.1.: Maritime Chartering Process Flow

2.1.2. Synthetic Data

The advancement of LLMs has increased interest in synthetic data generation as a means of mitigating data scarcity, preserving privacy, and reducing the costs of human annotation (Lu et al., 2023; Li et al., 2024a, 2023c). Synthetic generation also offers unprecedented scalability, enabling researchers and practitioners to generate datasets on demand without any data collection. Such scalability proves valuable when developing models that require exposure to edge cases that might be underrepresented in datasets, a technique known as data augmentation. However, training AI models exclusively on synthetic data carries risks, with model collapse representing a particularly significant concern. Model collapse refers to the progressive degradation of a model's performance resulting from training on artificially gener-

ated data (Shumailov et al., 2024). The collapse arises from the compounding of errors, which cause each successive model generation to learn from an increasingly simplified and less diverse dataset, eventually losing information about rare events. Furthermore, recent studies indicate that this collapse extends beyond diminished performance, as it also results in reduced diversity in the model's generated outputs (Guo et al., 2023). Crucially, to be considered reliable, artificially generated data must demonstrate faithfulness, characterized by logical and grammatical coherence, while simultaneously being diverse (Long et al., 2024).

2.1.2.1. Attribute prompting (AttrPrompting)

AttrPrompting is a synthetic data generation technique that instructs a language model to produce text with specified characteristics (Yu et al., 2024). These attributes can include aspects such as sentiment, style, topic, or structure, which guide the language model during text generation. Unlike purely unconstrained generation, AttrPrompting narrows the model's output space, thereby enabling the production of text that meets specific domain requirements. A key advantage of AttrPrompting lies in its flexibility. By tailoring prompts to particular keywords or contextual cues, researchers can generate text that accurately reflects the distribution of language features relevant for a given task. For instance, in the case of synthetic email generation, prompts can specify formal language, specialized jargon, or conversational patterns typical of corporate communication. This targeted control can help overcome data scarcity or privacy restrictions by producing training corpora that closely mimic real-world distributions without exposing sensitive content.

Another noteworthy finding from Yu et al. (2024) is that AttrPrompting is more cost-effective than simpler prompting strategies. Because the prompts are more specific, models require fewer attempts to produce texts that meet the desired specifications. In zero-shot settings, where no labeled data exists to guide prompt design via examples, AttrPrompting can still operate robustly if the attribute values are carefully selected or verified to avoid ambiguity. Yu et al. (2024) also show that targeted prompts can achieve comparable performance to larger, unstructured prompt sets at only a fraction of the computational costs. Hence, the method offers a twofold advantage: higher quality and diversity of generated text, as well as reduced overall generation cost compared to naive approaches.

Despite its benefits, AttrPrompting introduces several challenges that require careful consideration. Overly restrictive prompts may constrain textual diversity, pro-

ducing datasets that fail to capture the variability present in real-world corpora. Conversely, insufficiently detailed prompts might generate too varied content, making domain alignment during finetuning more difficult. The complexity of attribute selection amplifies this tension between specificity and diversity. Determining which attributes to specify and how to express them often requires substantial domain expertise. Moreover, when generating longer documents, such as emails, models may struggle to maintain consistent attribute expression throughout the entire text, resulting in coherence issues or attribute drift as the generation progresses. To ensure sufficient data quality, human-in-the-loop verification may be required, although this reintroduces manual overhead.

2.1.2.2. Base-Refine (BARE)

While AttrPrompting focuses on controlling specific text attributes, the BARE approach introduced by Zhu et al. (2025) takes a two-step strategy to balance diversity and quality in synthetic data generation. In the first step, a base model (often a general-purpose language model) produces content that is intentionally broad and varied, thereby maximizing diversity at the cost of lower quality. In the second step, an instruction-tuned model receives the output from the base model and refines it for clarity, consistency, and fluency without altering the overall content. According to the authors Zhu et al. (2025), this division of labor leverages the complementary strengths of both models. The base model excels at generating wide-ranging text, while the instruction-tuned model applies more stringent controls to ensure coherence and consistency. As a result, the final generation combines the benefits of both phases, featuring higher diversity than a purely instruction-tuned output and greater consistency than unrefined base-model generations.

BARE can be especially valuable in scenarios where domain specificity is paramount, but data availability is limited (Zhu et al., 2025). For example, in synthetic email generation, the base model can first introduce multiple stylistic variations or domain-specific entities. In contrast, the refined model imposes more rigorous constraints to maintain a coherent email-like structure. This two-step method can thereby produce a more representative and higher-quality corpus, mitigating some of the common drawbacks encountered in single-step synthetic data generation, such as repetitiveness, inconsistencies, or artificial phrasing. Zhu et al. (2025) demonstrate that even employing the relatively modest Llama-3.1-8B model (Grattafiori et al., 2024) as a base generator, followed by refinement using GPT4 (Achiam et al., 2023), can produce superior finetuning data compared to dynamic few-shot prompting.

Nonetheless, the BARE generation framework has its downsides. Firstly, it can lead to increased computational overhead and costs, especially in scenarios where both base and instruction-tuned models must be operated locally or sequentially. The empirical evidence presented by Zhu et al. (2025) suggests a positive correlation between model size and improvement in data quality, indicating that optimal results may require substantial computational resources. Furthermore, the efficacy of BARE demonstrates variability depending on the specific base-generator and refiner model combinations (Zhu et al., 2025), suggesting that researchers may need preliminary test runs to identify optimal configurations.

2.1.3. Text Mining

Text Mining is the process of uncovering insights from unstructured text (Hotho et al., 2005). A subfield relevant to this thesis is Information Extraction (IE), which aims to retrieve structured information from text (Allahyari et al., 2017).

2.1.3.1. Named Entity Recognition (NER)

Among IE, NER serves as a foundational component, as accurately identifying entities is a prerequisite for more complex extraction tasks. NER involves identifying and classifying entities into predefined categories such as persons, organizations, and locations (Nadeau and Sekine, 2007). Early NER systems relied on rule-based approaches, gazetteers, or handcrafted features (Mikheev et al., 1999), which were labor-intensive and did not account for semantic variations. The field subsequently evolved toward statistical machine learning methods such as Conditional Random Fields (CRFs) (Lafferty et al., 2001), which improved generalizability but still required extensive feature engineering.

The advent of deep learning has pushed the state-of-the-art performance of NER models by capturing contextual information more effectively (Li et al., 2020). Particularly transformer-based models like BERT (Devlin et al., 2019), have achieved breakthroughs by leveraging bidirectional context and attention mechanisms that dynamically weight the importance of surrounding words. Unlike previous approaches that processed text sequentially, transformers process entire sequences simultaneously, allowing the model to consider both left and right context when identifying entities. This contextual awareness enables models to disambiguate entities based on their context, for instance, distinguishing between *Apple* as a company versus a fruit. Finetuning pretrained transformer models on domain-specific NER tasks

has become standard practice, as it leverages the rich semantic knowledge acquired during pretraining on vast corpora (Li et al., 2020). However, these models typically require substantial labeled data to achieve optimal performance, which proves challenging in niche domains like maritime chartering, where annotated corpora are scarce.

2.1.3.2. Generative Information Extraction

A known limitation of traditional IE methods, such as NER, is their lack of generalizability. Once fine-tuned on a particular domain, the model is unlikely to perform well on domain shifts. Generative Information Extraction represents a paradigm shift that addresses these limitations by reformulating entity extraction as a text generation problem rather than a sequence labeling or classification task (Wang et al., 2023). This approach differs inherently from traditional methods in that it leverages the model's language generation capabilities to interpret and extract information simultaneously.

A survey by Zhang et al. (2025) reveals an increasing research interest in generative information extraction over the past few years. The generative approach offers several advantages for domain-specific applications, such as those related to maritime chartering emails. First, they demonstrate improved zero- and few-shot capabilities, thereby reducing the dependency on annotated datasets. Moreover, generative IE methods can capture semantic relationships between entities and perform implicit entity normalization, addressing the challenge of entity variations common in specialized domains (Zhang et al., 2025). Another key insight by Zhang et al. (2025) is the direct correlation between model scaling and extraction performance, with larger models demonstrating significantly improved adaptability and generalization capabilities. This so-called scaling law (Kaplan et al., 2020) has led to an emerging trend of model collaboration, where smaller pretrained language models (PLMs) work together with larger LLMs to balance computational efficiency and extraction quality. These collaborative approaches typically follow two primary patterns. In the first pattern, smaller PLMs perform the core extraction tasks. At the same time, LLMs serve in auxiliary roles, functioning as data generators or annotators that transfer knowledge into training datasets, or as discriminators and correctors that verify and refine extraction results from PLMs. For example, Zhang et al. (2025) highlight how LLMs can effectively generate supplementary training data to address scarcity issues or correct low-confidence extractions produced by smaller models. In the second pattern, larger LLMs perform the primary extraction while smaller PLMs provide task-specific knowledge.

GLiNER A notable generative extraction model is GLiNER (Generalist model for Named Entity Recognition) (Zaratiana et al., 2023), which addresses the drawbacks of both traditional NER models and LLM-based approaches. While traditional NER models are restricted to predefined entity types and LLMs require significant computational resources, GLiNER offers a compact yet flexible alternative. The architecture of GLiNER enables the parallel processing of entity types, rather than processing them sequentially, which significantly improves extraction efficiency. Despite being relatively compact (50M-300M parameters), GLiNER demonstrates superior zero-shot performance across diverse domains and languages, outperforming much larger finetuned models (Zaratiana et al., 2023).

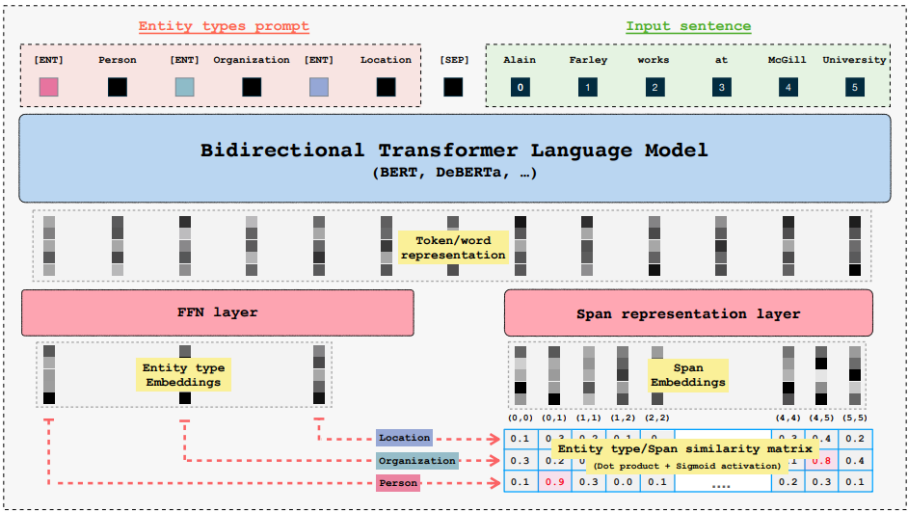

Figure 2.2.: The model processes a unified input sequence containing entity type prompts and an input sentence, encoded by a bidirectional transformer. It then computes embeddings for entity types and all candidate text spans through separate feed-forward networks. The core mechanism involves calculating a similarity score between each entity and span embedding via a dot product and sigmoid activation, effectively treating NER as a matching problem in a shared latent space. The example correctly identifies *Alain Farley* as a "Person" with a high matching score (0.9). Architecture from Zaratiana et al. (2023).

At its core, GLiNER consists of three components: a bidirectional transformer encoder (such as BERT (Devlin et al., 2019) or DeBERTa (He et al., 2021)), a span representation module, and an entity representation module (see Figure 2.2). The key innovation lies in reformulating NER as a matching problem in a shared latent space. Both the entity types to be extracted and the textual spans are processed through a unified embedding framework, enabling direct matching between them. GLiNER first constructs an input sequence by combining entity type prompts (e.g., *person*, *organization*) with the target text. Each entity type is preceded by a special [ENT] token, and the entire prompt section is separated from the text using a

[SEP] token. For example: "[ENT] person [ENT] organization [SEP] John works at Google." The bidirectional transformer then processes the entire sequence at once, allowing entity types and text to influence each other's representations through self-attention. For each entity type, GLiNER extracts the vector representation of its corresponding [ENT] token and passes it through a feedforward neural network to obtain refined entity embeddings. For the text segments, GLiNER considers all possible spans up to a maximum length (typically 12 tokens (Zaratiana et al., 2023)). Each potential span is represented by combining the embeddings of its boundary tokens (start and end positions) through another feedforward network. This creates span embeddings that capture the semantic content of each text fragment. The final matching step calculates compatibility scores between each entity type and each text span by computing the dot product of their respective embeddings, followed by a sigmoid activation function. This produces a probability score between 0 and 1, indicating whether a particular span belongs to a specific entity type. During inference, GLiNER selects spans with scores above a threshold (typically 0.5) as entity mentions, using a greedy selection algorithm to resolve overlapping spans (Zaratiana et al., 2023).

2.1.3.3. Template Filling

Template filling represents an alternative approach to IE that leverages the prompting capabilities of LLMs to extract structured information from unstructured text (Xu et al., 2024). Unlike traditional extraction methods, which identify and classify entities in isolation, template filling frames the extraction task as completing structured input prompts with relevant information from the source text. In some sense, template filling can be seen as the inverse of using AttrPrompting to generate synthetic data (Yu et al., 2024).

Recent advances have reformulated template filling as code generation rather than a natural language task (Li et al., 2023a, 2024b). This shift recognizes that templates align more closely with programming language constructs than with free-form text generation. Li et al. (2023a) demonstrated that code generation models could outperform text-based LLMs on information extraction tasks, leveraging the fact that code generation models are trained on vast amounts of structured data, making them naturally capable of producing outputs that conform to predefined schemas. An interesting approach is GoLLIE (Guideline-following Large Language Model for IE) (Sainz et al., 2023), which finetunes LLMs to comply with detailed annotation guidelines represented as Python docstrings and code comments. This methodology enables models to attend to nuanced entity descriptions and task-specific defini-

tions, rather than relying on pre-existing knowledge about entity types. It achieves promising zero-shot performance on unseen IE tasks.

2.2. Related Work

2.2.1. Synthetic Email Data

Academia offers limited in-depth investigations of artificially generated text for finance beyond data augmentation (Potluru et al., 2023), and research on email generation is notably sparse. Most publications around email generation address fraud detection scenarios, such as crafting adversarial samples to improve spam classifiers (Heiding et al., 2024; Karanjai, 2022). Synthesizing spam emails can be viewed as a form of data augmentation, as spam datasets often exhibit severe class imbalance towards legitimate messages, necessitating the creation of synthetic spam emails to balance the training data. Nevertheless, a recent study by Heiding et al. (2024) indicates that artificial spear phishing emails can be as deceptive as those written by humans, suggesting that content produced by LLMs can closely mirror real-world communication patterns.

Meanwhile, LLMs are increasingly being employed to assist with email composition, a trend that has been examined in multiple studies. For instance, Liu et al. (2022) investigated how humans perceive AI-generated emails. They found that while AI mediation can make communication more efficient, recipients may react negatively if they know the sender used an AI (Liu et al., 2022). Similarly, Li et al. (2025) conducted a systematic comparison of emails generated by LLMs and those written by humans. They found that AI-generated emails were more verbose, formal, and stylistically consistent while demonstrating lower linguistic diversity and predominantly neutral-to-positive sentiment. A qualitative evaluation with 41 participants who assessed emails without prior knowledge of authorship revealed that subjects preferred AI-generated responses but also considered them excessively verbose. Interestingly, a majority recognized AI authorship, especially that of generations by GPT4, with participants demonstrating an 82% detection accuracy. This observation contrasts with prior research on general text domains, where reliable discrimination between human and AI-generated content has proven challenging (Huang et al., 2025). One of the key challenges in synthesizing emails is injecting user profiles into the generation process, enabling LLMs to mimic individual writing styles and produce personalized content. This challenge was examined by Kumar et al. (2024) in their LongLaMP benchmark, which includes customized email completion as a core task. Using a

retrieval-augmented generation framework, they demonstrated that incorporating email histories improves personalization but might raise privacy risks as the model memorizes training data.

2.2.2. Text Mining on Emails

Email text mining constitutes a specialized domain within information extraction, presenting distinct analytical challenges. When researchers did tackle email mining, they quickly discovered fundamental differences from traditional text processing. Minkov et al. (2005) examined the extraction of personal names from emails and found that features successful in news articles proved far less effective in emails. In contrast to traditional documents, email text contains cross-referential dependencies across message threads. This requires processing individual messages in the context of the preceding dialogue. Additionally, email content exhibits high heterogeneity in terms of formality, structure, grammatical rigor, and length, ranging from brief, informal exchanges to comprehensive, formal business communications.

Despite the ubiquity of email communication in enterprise settings, surprisingly few studies have addressed text mining for email data. The lack of research can be attributed to the aforementioned privacy concerns and the resulting scarcity of publicly available data. When specialized approaches emerged, they were often based on heuristics. Laclavík et al. (2011) demonstrated that pattern-based extraction paired with company-specific gazetteers could achieve good performance. However, these approaches require customization for each enterprise and thus do not generalize.

Moreover, a comprehensive benchmark for email text mining emerged only recently with MAILEX (Srivastava et al., 2023), the first dataset designed for email event extraction. MAILEX contains 1,500 email threads from the ENRON corpus, with approximately 4,000 emails annotated with 8,000 event instances across 10 event types and 76 argument roles. The dataset follows email speech act theory, defining events through verb-noun pairs across three verb acts (Request, Deliver, Amend) and three noun acts (Data, Meeting, Action). This creates event types such as *Request Meeting*, *Deliver Data*, and *Amend Action*. The extraction task consists of two subtasks: trigger extraction, which identifies the minimal word spans that signal an event, and argument extraction, which identifies associated information, such as meeting attendees, dates, locations, or action descriptions. The dataset's evaluation revealed that even state-of-the-art models struggle with these challenges, with the best-performing BERT-based sequence labeling model achieving only 0.523 F1 for trigger classification and 0.368 F1 for argument classification, highlighting a

significant gap between email and traditional text extraction tasks.

2.2.3. Artificial Intelligence in Maritime Chartering

The interest of researchers in applying AI to maritime shipping has been growing rapidly over the past years (Xiao et al., 2024a; Munim et al., 2020; Filom et al., 2022). Publications predominantly address sustainability challenges, reflecting the industry's significant contribution to global greenhouse gas emissions (Durlik et al., 2024). Overall, text-mining applications within shipping remain scarce. Notable exceptions include Zhou et al. (2021) and Shin et al. (2018), who both employ hierarchical unsupervised Latent Dirichlet Allocation (LDA) techniques to analyze sustainability in shipping contexts, albeit with divergent objectives. While Zhou et al. (2021) analyze container shipping companies' sustainability reports, Shin et al. (2018) leverage LDA to conduct a bibliometric analysis of sustainability literature in maritime shipping.

The study that most closely resembles the scope of this thesis was conducted by Papageorgiou et al. (2024), who developed an email classification algorithm for shipbroking emails at a Greek agency. The authors note that the global shipbroker market, valued at $1.34 billion in 2021, relies heavily on efficient information management, making automated email classification systems essential for competitive advantage. Their research presents a two-step classification approach that first separates useful from non-useful emails and then sorts relevant emails into predefined categories with a combination of Term Frequency-Inverse Document Frequency (TF-IDF) vectorization and machine learning. Papageorgiou et al. (2024) highlight how the unstructured and acronym-rich nature of shipping communications creates significant challenges for automated classification. To overcome these challenges, their preprocessing pipeline incorporates dictionaries for maritime abbreviations and specialized tokenization rules. They also experimented with different vectorization techniques (Count Vectorizer, TF-IDF, and Word2Vec using both CBOW and Skip-gram architectures (Mikolov et al., 2013)), finding that TF-IDF consistently outperformed alternatives for capturing the semantic importance of shipping terminology. Overall, they achieved a classification accuracy of 99.6%. Unfortunately, neither the dataset nor the code was published.

3. Datasets

This chapter presents the collection of datasets used throughout this thesis, spanning from synthetic email generations to maritime corpora. Table 3.1 provides a high-level summary of all datasets and their primary application, before subsequent sections explore each resource in more detail.

Dataset	Source	Primary Use
Email Datasets		
Synthetic Email Chains	Generated (this work)	Training/evaluation of text mining models
Enron Corpus	Klimt and Yang (2004)	Genuine baseline for comparison
Maritime Domain Datasets		
Shipping Ports	Kaggle (Naik, 2023)	Port specifications for email generation and geocoding for freight rate calculation
Global Cargo Ships	Kaggle Ibrahim (2023)	Vessel specifications for email generation
Maritime News Articles	Web scraping (4 sources)	Domain knowledge for finetuning text mining models
Wikipedia Maritime	Web scraping	Maritime text corpus for pretraining and finetuning
IMO Vessel Codes	GitHub repository	Finetuning of text mining models
UN/LOCODE Ports	Web scraping	Finetuning of text mining models
General NLP Datasets		
Names Dataset	GitHub repository	Identity for email generation
NLI for SimCSE	Gao et al. (2021)	Contrastive pretraining
PileNER	Zhou et al. (2023)	NER model training
Template-Generated	Generated (this work)	NER finetuning
OOD NER Benchmark	Liu et al. (2021b)	Zero-shot NER evaluation
MTEB	Muennighoff et al. (2022)	Embedding evaluation

Table 3.1.: Overview of datasets used in this thesis, organized by category with their sources and primary applications.

The synthetic email chains serve as the primary evaluation target, while the maritime domain datasets provide the specialized knowledge necessary for generating realistic content and finetuning models. The general NLP datasets establish baselines and enable comparative evaluation against well-known benchmarks.

3.1. Email Chains

Email chains are the central component of this thesis, serving as both the primary dataset for generation and the evaluation target for text mining models. In this section, we detail the two corpora used: a synthetic dataset generated for this work and the Enron Corpus, which acts as a genuine baseline for comparison.

3.1.1. Synthetic Generations

The primary data source of this thesis are synthetically generated emails. In total, 19,817 email chains were generated using four distinct prompting strategies: Attr-Prompting (Yu et al., 2024), BARE (Zhu et al., 2025), few-shot prompting, and zero-shot prompting. Table 3.2 presents an overview of all prompting methodologies and their generation counts. The production goal was 1,000 email chains for attribute prompting and BARE methods, and 500 for few-shot and zero-shot approaches. Deviations from these targets reflect instances where generation models failed to produce the requested output format, resulting in invalid or incomplete email chains that were excluded from the final dataset. Detailed methodological procedures and generation parameters are discussed in Section 4. To illustrate the structure and content of these synthetic chains, a representative example of a liquefied natural gas (LNG) cargo negotiation is provided below.

Method	**Mistral**	**Deepseek**	**Claude**	**GPT4**	**Gemini**	**Row Total**
AttrPrompting	989	998	994	1000	1000	4,981
BARE (Llama-3.2-3B)	978	994	991	998	988	4,949
BARE (Llama-3.1-8B)	980	1000	993	997	994	4,964
Few-Shot	494	498	500	499	485	2,476
Zero-Shot	469	498	500	499	481	2,447
Total per Model	**3,910**	**3,988**	**3,978**	**3,993**	**3,948**	**19,817**

Table 3.2.: Synthetic email chain generation counts by model and methodology.

Example Email Chain: LNG Cargo Negotiation

Example Email Chain: LNG Cargo Negotiation

[2018-06-26 09:45]
From: will.turner@seawaybrokers.com
Subject: LNG Cargo - Jiangyin to Banjarmasin - MINERAL YANGFAN
Dear Jeffrey,
I trust this email finds you well. Following up on our earlier discussion, I'm pleased to present the following cargo opportunity:

Vessel:	MINERAL YANGFAN (206,392 DWT, LOA 300m)
Cargo:	1,819MT LNG
Load:	Jiangyin
Discharge:	Banjarmasin
Laycan:	15-20 July 2018
Freight Rate:	USD 107.69 PMT DDP
Demurrage:	USD 25,000 pdpr

Vessel's position is tight, and I recommend securing promptly if interested. Current market volatility makes me somewhat uncertain about maintaining this rate for long.
Best regards,
Will Turner

[2018-06-26 11:23]
From: j.mason@pacificcharter.com
Thank you for the offer. We're interested but find the rate somewhat high given current market conditions. We could offer full payment in advance if you can revise the rate to USD 98.50 PMT. This would eliminate any payment risk and provide immediate liquidity.

Please confirm vessel's holds are clean and suitable for LNG cargo, along with latest survey reports.
Best regards,
Jeffrey Mason

[2018-06-26 14:17]
From: will.turner@seawaybrokers.com
Dear Jeffrey,
Appreciate your prompt response and the prepayment offer. After consulting with owners, we can meet at USD 102.25 PMT with full advance payment, subject to stem confirmation. Vessel's holds are indeed clean, passed inspection last week, and fully LNG-ready. Will forward detailed survey reports separately.

Please note bbb (subject stem/sub details). Kindly advise if acceptable so we can proceed with the booking.
Best regards,
Will Turner

3.1.2. Enron Corpus

Originally released during the legal investigation of the Enron Corporation, the Enron Corpus (Klimt and Yang, 2004) has become a well-known benchmark in natural language processing (NLP) research. The corpus comprises over half a million email messages from 158 Enron employees, providing a rich variety of topics, writing styles, and recipient relationships. Researchers have utilized this dataset to explore authorship attribution (Fabien et al., 2020), thread summarization (Zhang et al., 2021), entity resolution (Dakle and Moldovan, 2020), and other applications. Moreover, the corpus is part of *The Pile*, an 800-gigabyte textual dataset used to pretrain language models (Gao et al., 2020).

In the context of this thesis, the Enron Corpus serves as the baseline for the synthetically generated emails, enabling meaningful comparisons between artificial and authentic email interactions. Benchmarking against a large-scale, human-authored email corpus enables the examination of how closely artificial emails resemble genuine conversation patterns and capture real-world diversity characteristics.

Example Email from Enron Corpus: Energy Crisis Alert

From: "Leonard, Alice" <ALeonard@caiso.com>
Sent: 12/13/2000 12:50 PM
To: ISO Market Participants
Subject: Important CAISO Notice: Possible firm load curtailments today
Be advised that there is a strong possibility that there will be 1000-3000MW of firm load curtailed this afternoon. In addition to ongoing issues related to gas prices, some participants are now indicating that they cannot sell into the markets due to credit concerns.

We are taking a number of actions to remedy the supply shortage, including work with State and Federal government representatives.

3.2. Datasets for Email Generation

To enhance the realism of the synthetically generated email chains, several external datasets were incorporated into the generation process. These resources contributed domain-specific language, realistic entity names, and plausible numerical values that one would expect to find in authentic maritime shipping negotiations.

3.2.1. Port Information

Port data was sourced from the *Shipping Ports Around The World* dataset on Kaggle (Naik, 2023), which contains over 450 major ports worldwide. These port locations were then geocoded to establish geographical coordinates, allowing for the calculation of shipping distances between the origin and destination ports. These distances, in turn, served as input parameters for a destination-based freight rate sampling algorithm, which generated plausible pricing structures for the synthetic email chains. During AttrPrompting, port names were provided as attributes for generating emails.

Port:	Amsterdam
Country:	Netherlands
UN Code:	NLAMS
Vessels in Port:	767
Vessel Movement:	Departures (24h): 179 — Arrivals (24h): 228
Type:	Port
Location:	North Sea (local), UK Coast & Atlantic (global)
Also known as:	NLAMS

3.2.2. Vessel Information

The *Global Cargo Ships Dataset* (Ibrahim, 2023) provides detailed vessel specifications, such as ship names, dimensions, and cargo-carrying capacities (quantified by deadweight tonnage). In the AttrPrompting framework, vessel specifications, including ship names, types, and capacity metrics, were utilized to generate emails containing technically accurate vessel details.

Ship Name:	MSC Loreto
Ship Type:	Container Ship
Built Year:	2023
Gross Tonnage (gt):	236184
Deadweight (dwt):	240000
Length:	399
Width:	60

3.2.3. Names Dataset

To simulate genuine sender and recipient identities, the *Name Dataset* (Remy, 2021) was used. Specifically, the 250 most common male and female first names, as well as the 400 most frequent surnames in the United States, were extracted from the dataset. During data generation, random pairings of these first names and last names were sampled to create realistic and diverse participant identities in the synthetic emails. For attribute prompting, these name combinations were provided as sender and recipient attributes, enabling the generation of over 100,000 unique identity pairings.

3.2.4. Maritime Chartering Datasets

To provide domain knowledge to the text mining models, textual data covering maritime chartering had to be collected from various sources. Unlike many established domains in NLP, there is no publicly available corpus for maritime chartering communications. This absence necessitated the creation of a custom corpus by aggregating data from multiple sources.

3.2.5. News Articles

Firstly, news articles were collected through daily scraping from March 7, 2025, to June 20, 2025, from four prominent maritime chartering news outlets. This process yielded over 5,000 distinct news articles as detailed in Table 3.2.5 and visually depicted as a word cloud in Figure 3.1.

News Source	**URL**	**Articles**
gCaptain	https://gcaptain.com	1,640
MarineTraffic	https://www.marinetraffic.com	491
Maritime Executive	https://maritime-executive.com	1,252
Splash247	https://splash247.com	1,993
Total		**5,376**

Table 3.3.: Maritime news sources and article counts collected through daily web scraping.

Figure 3.1.: Word Cloud of Collected News Articles

3.2.6. Wikipedia Data Collection

To build a text corpus for maritime chartering articles, we applied a snowball search strategy on the Wikipedia knowledge graph. Starting from manually selected seed articles, the system recursively discovered linked articles, thereby applying filtering mechanisms to prevent topic drift, such as distinguishing maritime vessels from medical blood vessels.

The collected articles then underwent a multi-stage preprocessing pipeline:

- **Text Cleaning**
  Removed coordinates, IPA pronunciation patterns, citations, and Wikipedia-specific formatting artifacts
- **Standardization**
  Normalized whitespace, quotes, and character encoding
- **Length Filtering**
  Retained texts between 10 and 1,200 words; larger texts were chunked
- **Maritime Relevance Assessment**
  Applied keyword-based scoring using a manually curated lexicon of maritime terms (vessel types, shipping terminology, trade expressions) to ensure domain-relatedness

Overall, this approach yielded a corpus of over 12,000 articles spanning ports, vessels, and chartering clauses, as visualized in Figure 3.2.

port
part
states
united
built
line
ships
service
city
boat
area
operation
world
war
island
system
route
company
time
vessels
canal
station
airport
water
aircraft
new
york
river
name
construction
university
university
university

Original Name	Cleaned Name	Type
MS TRUE	MS True	Oil/Chemical Tanker
S.ATLANTIC	Satlantic	Bulk Carrier
B.F.WELWATER	Bfwelwater	Bulk Carrier
HORIZON THEONI	Horizon Theoni	Oil/Chemical Tanker
YM VENU 8	Ym Venu 8	Oil/Chemical Tanker
CARTAGENA SUN	Cartagena Sun	Oil/Chemical Tanker
EVER-GREEN	Evergreen	Bulk Carrier
ATLANTIC II	Atlantic II	Bulk Carrier

Table 3.4.: Sample Vessel Names from Cleaned IMO Dataset

3.2.8. UN/LOCODE Code List by Country and Territory

Additionally, a dataset of ports was constructed by scraping location data from the United Nations Code for Trade and Transport Locations (UN/LOCODE) database (for Europe, 2025) and subsequent filtering for ports. The geographical distribution of the ports is depicted in Figure 3.3. We used the dataset to contrast ports and vessels within a (contrastive) finetuning objective for the text mining models.

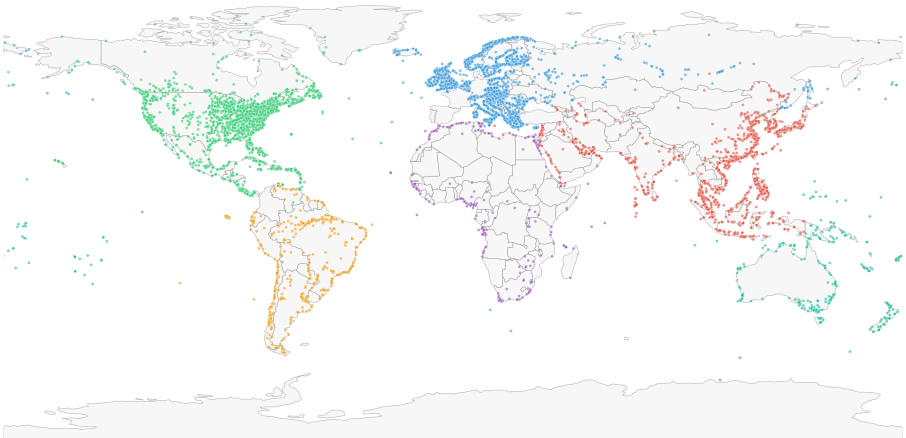

Figure 3.3.: Geographic distribution of over 10,000 ports derived from the UN/LOCODE database (for Europe, 2025). Each port is color-coded by its respective continent.

3.3. Natural Language Inference for SimCSE

Moreover, to test the impact of contrastive finetuning, the Natural Language Inference (NLI) dataset from (Gao et al., 2021) was used. This dataset contains sentence triplets, where the anchor sentence is semantically identical to one (called entailment or positive) and contradictory to the other (called contradiction or negative).

This triplet structure enables models to learn meaningful embeddings by pulling semantically similar sentences closer in vector space while pushing contradictory ones further apart. Below we show an example entry of the dataset:

Anchor:
my walkman broke so i'm upset now i just have to turn the stereo up real loud
Entailment (Positive):
I'm upset that my walkman broke and now I have to turn the stereo up really loud.
Contradiction (Negative):
My walkman still works as well as it always did.

3.4. NER Datasets

Last but not least, we employ three NER datasets: PileNER (Zhou et al., 2023) for general training, a template-generated maritime dataset for finetuning, and an Out-of-Distribution (OOD) NER Benchmark for zero-shot evaluation across diverse domains.

3.4.1. PileNER

PileNER (Zhou et al., 2023) is a large-scale NER dataset created by annotating 50,000 randomly sampled passages from the Pile corpus (Gao et al., 2020) via ChatGPT. The dataset exhibits a heavy-tailed distribution, with the top 1% of entity types accounting for 74% of total occurrences. This dataset serves as one of the primary instruction-tuning resources for universal NER models.

A known caveat of PileNER is the prevalence of rare labels that appear only once or very few times in the dataset (Zaratiana et al., 2023). To address this issue, we consolidated rare labels (appearing fewer than 10 times) with semantically similar, more frequent labels. The consolidation process involved:

1. Computing embeddings for all entity labels using the flag embedding model by Xiao et al. (2024b)
2. Calculating cosine similarities between rare and frequent labels
3. Mapping rare labels to (more) frequent ones when the similarity exceeded 0.8

As shown in Figure 3.4, this preprocessing improved label distribution. Singleton labels decreased from 4,153 to 2,487 (a 40% reduction), and unique entity types

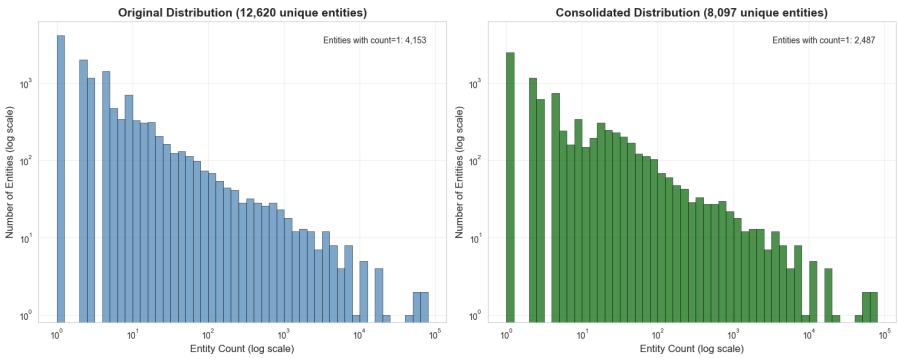

Figure 3.4.: PileNER label consolidation, showing a reduction in unique entities and singleton labels.

Rare Label	Mapped Label	Similarity
musicgroup	music group	0.976
duration of time	duration	0.921
dna base	genetic base	0.855
army	military	0.844
image program	image type	0.806

Table 3.5.: Examples of NER label consolidations using semantic similarity

decreased from 12,620 to 8,097 (a 36% reduction), resulting in a more balanced dataset for training. Some example mappings are presented in Table 3.5.

3.4.2. Template-Generated Maritime Training Data

To augment real-world data for finetuning, we developed a template-based generation system to create syntactically diverse maritime sentences with entity annotations. The system utilizes templates that range from brief operational messages to comprehensive market reports exceeding 1,000 characters, each targeting specific aspects of maritime communication. The templates, generated with Claude (Anthropic, 2024), are then populated using carefully selected entity lists, ensuring complete separation from the data used to create the email chains in the first place. Incoterms and commodities are manually curated lists, vessel names are drawn from the IMO dataset (Section 3.2.7), and ports are retrieved from UN/LOCODE (Section 3.2.8).

Template: The {commodity} shipment was sold {incoterm} {location} with buyer arranging onward transport.

Generated: The grain shipment was sold FOB Rotterdam with buyer arranging onward transport.

Annotations: ["grain <> commodity", "FOB <> incoterm", "Rotterdam <> location"]

3.4.3. Out-of-Distribution (OOD) NER Benchmark

This benchmark, used by (Zaratiana et al., 2023), integrates seven datasets to assess model generalization capabilities across specialized domains. Drawing from CrossNER (Liu et al., 2021b), it incorporates AI, Literature, Music, Politics, and Science domains, while also featuring Restaurant and Movie review datasets from MIT (Liu et al., 2013). This benchmark has become a frequently used test set for evaluating zero-shot NER performance. The composition of the dataset is presented in Table 3.6.

Domain	Samples	Entities	Unique Types	Top Entity Types
AI	431	1,624	13	task (219), field (207), product (197)
Literature	416	2,025	11	writer (566), book (418), literary genre (194)
Movie	2,442	5,338	12	genre (1,117), actor (812), year (720)
Music	465	3,163	12	musical artist (515), music genre (487), band (462)
Politics	650	3,943	8	political party (953), location (599), organization (513)
Restaurant	1,520	3,150	8	location (812), amenity (533), cuisine (532)
Science	543	2,568	16	scientist (471), astronomical object (337), organization (273)
Total	**6,467**	**21,811**		

Table 3.6.: Statistics of the OOD NER Benchmark showing sample counts, total entities, unique entity types, and the three most frequent entity types per domain.

3.5. Massive Text Embedding Benchmark (MTEB)

To evaluate the effectiveness of different embedding finetuning approaches, we use the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2022). MTEB is an evaluation framework spanning eight embedding tasks across 58 datasets and 112 languages, providing standardized metrics for assessing text embedding quality across diverse applications. For this thesis, a representative subset of MTEB tasks was selected. This selection was chosen to strike a balance between computational efficiency and comprehensive evaluation.

- **Semantic Textual Similarity (STS)**
  Six datasets including *BIOSSES*, *SICK-R*, *STS12*, *STS16*, *STSBenchmark*,

and *STS22*, measuring how well models capture semantic similarity between sentence pairs

- **Classification**
  Banking77 and *AmazonCounterfactual* datasets, evaluating embeddings' effectiveness for downstream classification tasks

- **Clustering**
  TwentyNewsgroups dataset, assessing how well embeddings group semantically related texts

- **Retrieval**
  FiQA2018 dataset, testing information retrieval capabilities

- **Reranking**
  AskUbuntuDupQuestions dataset, evaluating the ability to rank relevant documents

4. Methodology

This chapter outlines the methodology underpinning this thesis. The first part explains the approach to synthetic data generation, explaining the distinct prompting strategies used to create the maritime email corpus. We then elaborate on the metrics used to evaluate the quality of the resulting data, including textual diversity, sentiment, verbosity, and authorship detectability. The second part transitions from data creation to its application, presenting the text mining approaches developed for maritime entity extraction. Finally, the chapter concludes by describing the analytical techniques used to evaluate the quality of the learned entity embeddings.

4.1. Synthetic Data

This section introduces the methodology for generating the email data, followed by a discussion of the metrics used to evaluate the resulting email chains.

4.1.1. Data Generation Methods

The quality of synthetic training data fundamentally determines the performance of downstream text mining models. Maritime chartering emails present unique challenges: they combine technical terminology, negotiation dynamics, and industry-specific jargon that general-purpose LLMs may not capture accurately. We therefore develop four complementary generation strategies to assess the viability and robustness of LLMs for email generation.

4.1.1.1. Attribute Prompting (AttrPrompting)

To generate realistic and diverse email chains under AttrPrompting, we developed a comprehensive dictionary of attributes that captures maritime elements and stylistic nuances. These include vessel specifications such as weight and length metrics, port names, cargo sizes, and commodities. In addition, the dictionary contains

meta-instructions for email synthesis, guiding the model on verbosity, syntactic variations, English proficiency, and probabilistic anomalies such as typographical errors or incorrect information that gets corrected in follow-up emails. At the start of each synthetic chain, we randomly select values for these attributes, which are then embedded into a prompt template (see Appendix A.1.2), resulting in several billion unique prompt configurations. In Appendix A.1.1, we provide a complete description of all attributes and their possible values.

In addition to these attributes, each port was geocoded into latitude and longitude coordinates using GeoPandas (Jordahl et al., 2020). From these coordinates, we built a distance matrix using the Haversine formula, which calculates distances between points on a sphere. While this provides a reasonable approximation for voyage planning, actual shipping routes are constrained by coastlines, canals, and maritime regulations—factors not captured by direct geodesic distances:

$$d = 2\,r\,\arcsin\Bigg(\sqrt{\sin^2\!\Big(\frac{\Delta\varphi}{2}\Big) \;+\; \cos(\varphi_1)\,\cos(\varphi_2)\,\sin^2\!\Big(\frac{\Delta\lambda}{2}\Big)}\Bigg), \tag{4.1}$$

where $\Delta\varphi$ and $\Delta\lambda$ are the differences in latitude and longitude (in radians), and $r = 6371$ is Earth's approximate radius in kilometers. These distances underpin a distance-based freight rate computation engine, which applies an initial base rate of 30 and increments it by a random step-up factor for every additional 1,000 km in distance. To provide context for the computed rate, a currency is chosen at random. This freight quote then acts as the starting point for negotiations between the broker and the charterer.

4.1.1.2. Base-Refine (BARE)

The BARE methodology requires high-quality seed emails to guide the base model's generation process, unlike AttrPrompting, which relies on structured attributes. BARE leverages example-driven generation, where the quality and diversity of initial examples directly influence the output. To establish this foundation, we created a curated corpus of 100 seed emails that capture the essential characteristics of maritime chartering communications. The seed emails are shown in Appendix A.2.1.

These seed emails were initially generated using ChatGPT with prompts designed to elicit diverse shipping scenarios, including vessel negotiations, port discussions, cargo specifications, and rate negotiations. Each generated email was then manually reviewed and adjusted to ensure realistic use of maritime terminology and abbrevi-

ations, varied business tone and negotiation dynamics, diverse entity distributions across vessels, ports, and commodities, and varied email lengths and styles.

Base Generation In the first stage of BARE, we exploit the output diversity provided by underlying base models (Zhu et al., 2025). As part of this research, both Llama-3.1-8B and Llama-3.2-3B (Grattafiori et al., 2024) were hosted locally using the vLLM library (Kwon et al., 2023), which enables distributed serving of Hugging Face models via tensor parallelism (Kwon et al., 2023). The model instances ran on two NVIDIA T4-16GB GPUs. To facilitate user interaction, each model was wrapped into a FastAPI (Ramírez, 2025) application and exposed through a generate endpoint, allowing clients to post prompts. On average, both local model instances operated at a decoding rate of approximately 20 tokens per second, processing inputs at a rate of around 100 tokens per second.

To initiate a new email chain, we first generate a starting email using a few-shot approach. Two representative emails, randomly sampled from the 100 shipping-related seed emails, are incorporated into a prompt to encourage the model to generate a new message with a similar style. A prompt example for the base generation can be found in Appendix A.2.2.1.

Once the starting email candidate is obtained, our implementation iteratively prompts the LLM with the progressively growing email chain to generate subsequent emails. This process is repeated a random number of times (between 2 and 5) to produce email threads of variable length. Each turn references the existing chain to maintain consistency in topic and style. To control the scope of the generation, a placeholder (for example, `<Email i>:`) is appended to the prompt, guiding the model to generate precisely one response at a time. The response is then clipped at the next placeholder, ensuring that each model invocation yields only a single email segment. A complete example of this iterative prompt can be found in Appendix A.2.2.2.

Refinement After obtaining base-generated email threads, each chain is refined for coherence, correctness, and stylistic consistency using the same LLMs employed in the AttrPrompting approach. In line with Zhu et al. (2025), the refinement model is explicitly instructed to retain the original tone and theme of the email. The refinement prompt is shown in the Appendix A.2.3.1. Post-generation refinement of the entire chain was selected primarily for cost efficiency, as batch processing API requests reduces per-token costs by approximately 50%. To validate this methodological choice, we performed iterative refinement with Mistral (Jiang et al., 2024)

and Claude (Anthropic, 2024). In this alternative approach, each email was refined immediately after its base generation, and the refined version was then fed back into the base model for subsequent messages. As demonstrated in the ablation study in Section 5.1.5, this iterative approach yielded unfavorable results.

4.1.1.3. Zero-Shot Prompting

To establish a baseline against which the other generation methods can be evaluated, we also employ zero-shot prompting. In zero-shot prompting, no prior examples or domain-specific context are provided to the model; instead, it receives a single prompt that instructs it to generate an email chain from scratch. This stands in contrast to AttrPrompting and BARE, where a set of parameters or examples guides the generation. The model is thus left to infer or invent these details on its own. By comparing outputs produced under zero-shot prompting to those generated through AttrPrompting and BARE, we can discern how much of the outcome can be attributed to carefully engineered prompts and contextual clues. The zero-shot prompt used can be found in the Appendix A.3.1.

4.1.1.4. Few-Shot Prompting

Finally, few-shot prompting offers an intermediate approach between zero-shot prompting and methods such as AttrPrompting. In this thesis, we implemented few-shot prompting by selecting three example chains from a manually curated set of 20 email chains. These 20 chains were drawn from each of the five examined models under AttrPrompting and BARE (four chains per model). They were chosen to cover a range of shipping scenarios, negotiation styles, and linguistic patterns. By presenting these examples, the language model gains a clearer understanding of the desired content and tone, enabling it to generate new email chains more effectively. The full few-shot prompt is provided in the Appendix A.4.

4.1.2. Evaluation and Analysis

Validating synthetic data quality requires metrics that capture both surface-level linguistic properties and deeper semantics. We therefore employ a multifaceted evaluation framework that examines textual diversity (to prevent model collapse), sentiment patterns (to ensure a realistic business tone), verbosity (to match industry communication styles), and authorship detectability (to quantify human-likeness). To maintain consistency across various data generation methods, 450 samples were

randomly selected from each model and method for diversity analysis. These subsets were then benchmarked against a 450-email subsample from the human-authored Enron corpus.

4.1.2.1. Textual Diversity

Lexical Diversity To measure the variety of vocabulary in the generated email chains, we adopt the distinct-n metrics introduced by Li et al. (2015). Although the Type-Token Ratio (TTR) (Templin, 1957) is a traditional measure of lexical diversity, it is susceptible to text length (Shaib et al., 2024), making it less suitable for the comparatively short texts typical of email communications. In contrast, distinct-n offers a robust alternative by assessing the ratio of unique n-grams to the total number of tokens. Formally, the distinct-n metric is defined as follows:

$$\text{Distinct-n} = \frac{\text{number of unique n-grams}}{\text{total number of tokens}} \tag{4.2}$$

Three variants of the metric are examined in this thesis: distinct-1 (unigrams), distinct-2 (bigrams), and distinct-3 (trigrams). Distinct-1 captures word-level diversity, quantifying the proportion of unique words in the generated text. Distinct-2 measures phrasal diversity by quantifying the variety of word pairs, and distinct-3 extends this measurement to three-word sequences, offering insights into the model's ability to avoid repetitive patterns. Together, these metrics provide a comprehensive assessment for identifying repetitiveness in generated text, a known challenge in neural text generation systems (Chen et al., 2024; Padmakumar and He, 2023).

Semantic Diversity In addition to the statistical variety captured by lexical metrics, semantic diversity evaluates the breadth of conceptual coverage in the generated emails, quantifying the generation model's ability to span diverse domains. To measure diversity at a semantic level, we use the sentence-transformers/all-MiniLM-L6-v2 model (Reimers and Gurevych, 2019) to generate vector embeddings of email texts. The methodology involved:

1. **Text Preprocessing**:
   Converting texts to lowercase, normalizing whitespace, and preserving decimal notation to ensure consistent encoding.

2. **Semantic Encoding**:
   Generating fixed-dimensional vector representations for each email using the transformer model's [CLS] token embedding.

3. **Similarity Computation**:
   Calculating pairwise cosine similarity between all email embeddings to quantify semantic overlap.

Semantic diversity was then defined as the complement of the average pairwise similarity:

$$\text{Semantic Diversity} = 1 - \frac{1}{n(n-1)/2} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \text{sim}(e_i, e_j) \tag{4.3}$$

where n is the number of emails, e_i and e_j are the embeddings of the i-th and j-th email, and $\text{sim}(\cdot, \cdot)$ denotes the cosine similarity. Higher semantic diversity values indicate greater dispersion and broader topic coverage.

The measurement of semantic diversity was limited to the first email of each chain. This choice is grounded in the observation that the initial message establishes the thematic scope of the conversation. By embedding only the first email, the analysis isolates the model's capacity to initiate a wide range of potential conversational contexts. Furthermore, examining only the initial email minimizes any overlap of themes influenced by later exchanges and is computationally more efficient.

4.1.2.2. Sentiment

Furthermore, to examine whether the generation method influences the sentiment of the generated email chains, a dual-model ensemble approach was implemented to mitigate model-specific biases.

Text Preprocessing Before sentiment computation, all email texts went through a preprocessing pipeline that removed extraneous elements that could confound sentiment assessment. This preprocessing involved the removal of HTML tags, normalization of whitespace and line breaks, and elimination of non-content elements, including disclaimers, promotional content, and forwarding headers. Additionally, quoted message history was discarded to ensure that only the primary content of each email chain was considered in the analysis.

Ensemble Model Architecture The sentiment analysis framework employed an ensemble approach, combining two models: Flair (Akbik et al., 2019) and RoBERTa-large (Hartmann et al., 2023), with equal weights (0.5) assigned to both models'

predictions. To accommodate token limitations, texts exceeding the maximum sequence length (512 tokens) were segmented into smaller chunks, with sentiment computed for each chunk individually before aggregation. The sentiment classification followed a probabilistic framework with defined thresholds, which were translated into normalized probabilities for three sentiment classes (negative, neutral, positive) through a piecewise mapping function.

$$P_{negative} = \begin{cases} 1.0 - P_{ensemble}, & \text{if } P_{ensemble} < 0.4 \\ 0.6 - P_{ensemble}, & \text{if } 0.4 \leq P_{ensemble} < 0.45 \\ \max(0, 0.4 - 0.5 \cdot P_{ensemble}), & \text{if } P_{ensemble} \geq 0.45 \end{cases} \tag{4.4}$$

$$P_{positive} = \begin{cases} P_{ensemble}, & \text{if } P_{ensemble} > 0.6 \\ P_{ensemble} - 0.1, & \text{if } 0.55 < P_{ensemble} \leq 0.6 \\ \max(0, 0.5 \cdot P_{ensemble} - 0.1), & \text{if } P_{ensemble} \leq 0.55 \end{cases} \tag{4.5}$$

$$P_{neutral} = 1.0 - (P_{neg} + P_{pos}) \tag{4.6}$$

Chain-Level Analysis The pipeline aggregated preprocessed texts from all messages within an email chain and then applied the sentiment analyzer to the concatenated text. By doing so, it determines the overall sentiment of the entire conversation rather than calculating a separate score for each message.

4.1.2.3. Verbosity

Email verbosity was measured using the NLTK tokenizer (Bird et al., 2009) to compute three metrics: (1) words per email, (2) sentences per email, and (3) words per sentence. These metrics were calculated as the total word count divided by the number of emails, the total sentence count divided by the number of emails, and the total word count divided by the total sentence count, respectively.

4.1.3. Authorship Detection

To conclude the assessment of the synthetic data, we apply DetectGPT (Mitchell et al., 2023) for authorship detection. In line with the original paper, 15% of each email is masked with a span length of three words and then regenerated using T5-3B (Raffel et al., 2020). In total, each email is paraphrased 20 times to establish a

robust comparative baseline. Then, both original and rewritten emails are evaluated for their negative log-likelihood under GPT-2 XL (Radford et al., 2019). Finally, the DetectGPT score is calculated by comparing the likelihood of the original text against the average probability of its perturbed variants.

Formally, given ℓ_{orig} as the negative log-likelihood of the original email and ℓ_i as the negative log-likelihood of the i-th perturbed version, the perturbation score can be computed using a log-ratio:

$$\log(\ell_{\text{orig}}) - \log\left(\frac{1}{N}\sum_{i=1}^{N}\ell_i\right) \tag{4.7}$$

The efficiency of DetectGPT is based on a fundamental distinction in text generation between humans and machines. Since LLMs are trained to maximize the probability of the next token, machine-generated text typically occupies the peak of the likelihood distribution compared to its perturbations (Mitchell et al., 2023). In contrast, the human-authored text does not exhibit this maximization paradigm. It thus demonstrates more variability concerning the perturbations, meaning that paraphrased versions can score higher or lower than the original version (Mitchell et al., 2023). This asymmetry in perturbation sensitivity provides the discriminative methodology that underlies DetectGPT, where higher scores indicate synthetic generation and lower scores suggest human authorship (Mitchell et al., 2023).

Human-written emails are drawn from a randomly sampled subset of the ENRON corpus introduced in Section 3.1.2. Each email is preprocessed to remove disclaimers, promotional material, forwarding headers, attachments, and meeting invitations. Finally, all processed texts exceeding 150 words are truncated to mitigate computational overhead and to adhere to the input limitation of the paraphrasing model T5.

4.2. Text Mining

Having generated synthetic maritime emails, we now address the core challenge of extracting structured information from these unstructured communications. Maritime entity extraction presents unique difficulties: vessel names can overlap with locations (e.g., *MV Singapore*), commodities require contextual disambiguation, and Incoterms can be obfuscated in maritime abbreviations. This section presents the methodology underlying three distinct approaches for maritime entity extraction,

ranging from statistical methods to neural architectures. We begin with a TF-IDF baseline that leverages domain-specific patterns without requiring training, followed by a comprehensive exploration of GLiNER-based models with various pretraining and finetuning strategies, and conclude with parameter-efficient finetuning (PEFT) of the compact Qwen-0.5B language model.

4.2.1. Evaluation Metric: Macro F1

All models are evaluated using the macro F1 score, which assigns equal weight to each entity type, regardless of class frequency. We adopt macro F1 as the primary evaluation metric for several reasons:

1. **Entity Type Imbalance**: Maritime datasets exhibit significant imbalance across entity types. Vessels and locations appear frequently (often multiple per email), while Incoterms typically occur at most once per email chain. Macro averaging prevents high-frequency entities from dominating the overall score.
2. **Equal Importance**: In maritime operations, accurate extraction of all entity types is critical. Missing an Incoterm can have legal ramifications as severe as misidentifying a vessel. Macro F1 reflects this by assigning equal weights to each entity type.
3. **Model Comparison**: When comparing models, macro F1 reveals performance differences across all entity types, whereas micro F1 might mask poor performance on rare entities.

We also compute micro-averaged precision and recall to provide complementary insights into model behavior. Micro metrics aggregate all entity-level decisions across types, which can be more informative when analyzing extraction patterns in datasets where entity co-occurrence varies.

4.2.1.1. Precision, Recall, and F1 Score

For each entity type $e \in \{\text{vessel}, \text{port}, \text{commodity}, \text{incoterm}\}$ and each sample i, we calculate:

$$\text{Precision}_{e,i} = \frac{\text{TP}_{e,i}}{\text{TP}_{e,i} + \text{FP}_{e,i}} \tag{4.8}$$

$$\text{Recall}_{e,i} = \frac{\text{TP}_{e,i}}{\text{TP}_{e,i} + \text{FN}_{e,i}} \tag{4.9}$$

$$\text{F1}_{e,i} = 2 \cdot \frac{\text{Precision}_{e,i} \cdot \text{Recall}_{e,i}}{\text{Precision}_{e,i} + \text{Recall}_{e,i}} \tag{4.10}$$

where TP, FP, and FN represent true positives, false positives, and false negatives, respectively. Our evaluation employs substring matching to account for minor variations in entity representation (e.g., *MV ATLANTIC* matching *atlantic*).

4.2.2. TF-IDF Extraction

Before deploying complex neural models, we first establish whether simpler statistical methods can adequately capture maritime entities. TF-IDF offers several advantages for this domain: it requires no training data (crucial when real emails are scarce), provides interpretable results through term importance scores, and can be rapidly adapted to new entity types by modifying pattern rules. This baseline helps quantify the performance of more sophisticated approaches.

Our extraction system uses spaCy (Honnibal et al., 2020) for preprocessing and scikit-learn's TF-IDF vectorizer (Pedregosa et al., 2011) for term importance analysis. The configuration parameters used for the TF-IDF vectorizer are outlined in Table 4.1:

Parameter	**Value**	**Parameter**	**Value**
Max features	5,000	Min document frequency	2
N-gram range	(1, 3)	Max document frequency	0.8
Token pattern	`[a-zA-Z][a-zA-Z0-9/-]*`	TF-IDF threshold	0.1

Table 4.1.: TF-IDF Vectorizer Configuration for Domain Term Identification

The extraction methodology employs entity-specific pattern-matching rules augmented with domain knowledge. For each entity type, we define indicator terms and contextual patterns as shown in Table 4.2:

The extraction pipeline follows a multi-stage approach:

1. **Corpus Analysis**: Initialize TF-IDF on the combined corpus from all synthetic email datasets to identify domain-specific high-importance terms

Entity Type	Indicator Terms and Patterns
Vessel	`mv`, `m/v`, `mt`, `m/t`, `ss`, `s/s`, `bulk carrier`, `tanker`, `vessel`, `ship`
Port/Location	`port`, `terminal`, `berth`, `from`, `to`, `load`, `discharge`, `destination`, `origin`
Commodity	`cargo`, `commodity`, `tons`, `mt`, `tonnes`, `barrels`, `shipment` + quantity patterns
Incoterm	Three-letter uppercase patterns with contextual terms: `terms`, `basis`, `delivered`, `freight`

Table 4.2.: Entity-Specific Indicators and Extraction Patterns

2. **Pattern Matching**: Apply regular expressions tailored to maritime terminology:
   - Vessel patterns:
     `(?:vessel_prefix)\s+([A-Z][A-Za-z0-9\s\-\.]{2,30})`
   - Port patterns:
     `(?:port_indicator)\s+(?:of|at|in)?([A-Z][a-zA-Z\s\-]{2,30})`
   - Commodity patterns with quantities:
     `(\d+(?:,\d+)?)\s*(?:units)\s+(?:of)?([a-zA-Z]+)`
3. **Named Entity Recognition**: Leverage spaCy's pretrained NER to identify organizations (vessels), locations (ports), and products (commodities) with contextual validation
4. **Domain-Term Matching**: Cross-reference extracted candidates with high-TF-IDF terms identified during corpus analysis

The system processes email threads by concatenating individual messages and applying the extraction pipeline to the complete text. Unlike finetuned approaches, it requires no training time and can be immediately applied to new domains by modifying the indicator terms and patterns.

4.2.3. GLiNER-based NER Models

While the TF-IDF baseline provides interpretable results without training, maritime communications contain complex contextual patterns that rule-based systems might struggle to capture. GLiNER's architecture (Zaratiana et al., 2023) offers a compelling middle ground: it can leverage pretrained language understanding to handle arbitrary entity types without architectural changes. This makes it ideal

for the evolving terminology in maritime chartering. This section presents a two-stage training pipeline for maritime NER. Our approach systematically examines how various encoder pretraining strategies impact the downstream extraction performance of GLiNER. All experiments were conducted on two NVIDIA T4 GPUs, each equipped with 16 GB of VRAM.

- **Stage 1: Encoder Pretraining**
  First, five different variants of DeBERTa-v3-small (He et al., 2021) are created through different pretraining objectives.
- **Stage 2: GLiNER Integration and finetuning**
  Then, we integrate these pretrained encoders into the GLiNER architecture as the backbone model (used to encode text spans and entity labels), train them on the PileNER dataset (see Section 3.4.1), and apply four different finetuning strategies.

Overall, this approach produces 20 model variants (5 encoder types × 4 finetuning strategies) for evaluation. The complete training pipeline required approximately 20 hours per model variant. Figure 4.1 illustrates the training pipeline at a high level.

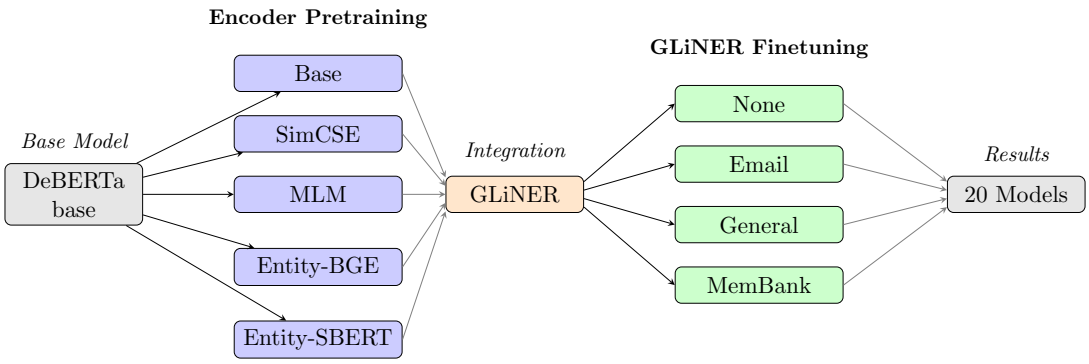

Figure 4.1.: GLiNER training pipeline showing the flow from base model through encoder pretraining and GLiNER finetuning to final model variants.

4.2.3.1. Stage 1: Encoder Pretraining Strategies

Before integrating encoders into GLiNER, we explore different strategies for preparing DeBERTa-v3-small encoders (He et al., 2021). These include a baseline approach using the original pretrained weights without additional tweaking, as well as four pretraining strategies designed to enhance different aspects of entity recognition: entity disambiguation, sentence-level understanding, and domain adaptation.

Contrastive Pretraining for Entity Disambiguation The maritime domain presents unique ambiguity challenges for NER models, particularly between vessels and locations. Vessel names such as *MTM Singapore*[3] exemplify how entity type boundaries can be blurred. Building upon insights from Genest et al. (2025), who demonstrated that contrastive finetuning can enhance open-domain NER performance; we develop two entity-contrastive variants using different similarity metrics: BGE embeddings (Xiao et al., 2024b) and Sentence-BERT (Reimers and Gurevych, 2019).

Our contrastive framework modifies DeBERTa-v3-small by adding a projection head that reduces 768-dimensional hidden states to 256-dimensional embeddings. The projection head in contrastive learning was initially proposed in computer vision by Chen et al. (2020) and serves a dual purpose. It forces the model to learn non-linear feature interactions while also preventing overfitting to the contrastive objective. This approach was successfully adapted to sentence embeddings by Gao et al. (2021), who found that using projection heads during training and then discarding them at inference time can lead to improved performance.

To facilitate entity contrasting, we implement a stratified batch sampling strategy that leverages semantic relationships between entity types. All entity type labels are encoded using Sentence-BERT (all-MiniLM-L6-v2) (Reimers and Gurevych, 2019) and BGE Embedding (Xiao et al., 2024b) to compute a pairwise cosine similarity matrix, which is then partitioned into four distinct strata as shown in Table 4.3.

Similarity Range	**Description**	**Example Pair**
0.7–1.0	Near-synonymous types	Vessel ↔ Ship
0.5–0.7	Related but distinct	Port ↔ Terminal
0.2–0.5	Loosely related	Vessel ↔ Company
0.0–0.2	Semantically distant	Vessel ↔ Date

Table 4.3.: Similarity Strata for Stratified Batch Sampling

To balance positive alignment signals with hard negative discrimination, a structured batch composition is applied. Each 64-sample batch is constructed according to the distribution shown in Table 4.4, ensuring both semantic coherence and sufficient diversity for effective contrastive learning.

To further exploit the semantic relationships captured in the cosine-similarity matrix of entity labels, we apply a modified version of the InfoNCE loss (Oord et al., 2018). Traditional contrastive learning approaches treat negative pairs in a binary

[3] https://www.marinetraffic.com/en/ais/details/ships/shipid:689144/mmsi:563117800/imo:9477529/vessel:MTM_SINGAPORE

Sample Type	Count	Types	Similarity Stratum
Anchor	16	1	Selected entity type
High similarity	8	1	0.7–1.0
Medium similarity	6	2	0.5–0.7
Low/Very low similarity	8	4	0.0–0.5
Adaptive filling	Variable	Variable	Any stratum

Table 4.4.: Batch Composition for Stratified Contrastive Sampling

fashion, meaning that samples are either positive matches or negative non-matches. However, this binary treatment fails to capture the relationships that exist between different entity types. Our modified loss function replaces these hard binary labels with continuous similarity scores derived from the cosine-similarity matrix of the encoded entity labels. This design choice reflects an important insight: not all misclassifications are equally problematic. The similarity matrix captures these gradations, allowing the model to understand that confusing semantically related entities (e.g., labeling a "Vessel" as a "Ship") is less severe than fundamental category mistakes (e.g., labeling a "Vessel" as a "Person"). This leads to the following InfoNCE loss function:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} p_{ij} \log \frac{\exp(s_{ij}/\tau)}{\sum_{k=1}^{N}\exp(s_{ik}/\tau)} \tag{4.11}$$

where:

- $s_{ij} = \cos(\mathbf{e}_i, \mathbf{e}_j)$ measures the cosine similarity between embeddings $\mathbf{e}_i$ and $\mathbf{e}_j$
- $p_{ij} \in [0, 1]$ represents the soft target similarity between entity types i and j, derived from the precomputed similarity matrix
- $\tau = 0.25$ is the temperature parameter that controls the concentration of the similarity distribution

While the InfoNCE loss encourages embeddings to align with the target similarity structure, it can lead to representation collapse where embeddings cluster in a small region of the embedding space. To counteract this and ensure effective utilization of the representation space, we incorporate a uniformity regularization term following Wang and Isola (2020):

$$\mathcal{L}_{\text{uniform}} = \log \frac{1}{N(N-1)} \sum_{i \neq j} \exp(-\|\mathbf{z}_i - \mathbf{z}_j\|_2^2) \tag{4.12}$$

This loss function promotes uniform distribution of normalized embeddings $\mathbf{z}_i = \mathbf{e}_i/\|\mathbf{e}_i\|$ on the unit hypersphere by penalizing configurations where embeddings are too close together (Wang and Isola, 2020). The exponential term $\exp(-\|\mathbf{z}_i - \mathbf{z}_j\|_2^2)$ assigns higher penalties to nearby embedding pairs. By minimizing this average pairwise similarity, the uniformity loss encourages embeddings to spread out. Our final training objective combines both the modified InfoNCE loss and the regularization term and thus becomes:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{InfoNCE}} + \lambda \mathcal{L}_{\text{uniform}} \tag{4.13}$$

where $\lambda = 0.05$ balances the contribution of the uniformity term.

The model was trained with the AdamW optimizer (Loshchilov and Hutter, 2017) on the PileNER dataset labels (see Section 3.4.1) with the hyperparameters outlined in Table 4.5:

Parameter	**Value**	**Parameter**	**Value**
Base Model	DeBERTa-v3-small	Learning Rate	5×10^{-6}
Temperature (τ)	0.25	Batch Size	64
Alignment Weight	1.0	Anchor Samples	16
Uniformity Weight	0.05	Epochs	3
Projection Dim	256	Gradient Clip	0.5
Mixed Precision	FP16		

Table 4.5.: Entity-Contrastive Training Hyperparameters

SimCSE Pretraining for Sentence-Level Understanding To complement the entity-specific contrastive learning approach, we also explore supervised SimCSE (Simple Contrastive Learning of Sentence Embeddings) (Gao et al., 2021). Different from entity-level contrasting, SimCSE operates at the sentence level, leveraging semantically related sentence triplets. The method uses entailment pairs as positive examples and contradiction pairs as hard negatives. During training, the embeddings of the target sentence (also referred to as the anchor) and its entailment are pulled closer together in the semantic space, while the contradiction is pushed apart. This contrastive mechanism ensures that semantically similar sentences cluster together. The training objective uses the InfoNCE loss (Oord et al., 2018) with temperature scaling to learn discriminative sentence embeddings. Given a batch of sentence triplets $(s_i^{anchor}, s_i^{positive}, s_i^{negative})$, the model computes embeddings $\mathbf{e}_i^{anchor}$, $\mathbf{e}_i^{positive}$, and $\mathbf{e}_i^{negative}$ through the encoder and pooling mechanism. The InfoNCE loss is then computed as:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\text{sim}(\mathbf{e}_i^{\text{anchor}}, \mathbf{e}_i^{\text{positive}})/\tau)}{\mathcal{Z}_i} \tag{4.14}$$

where $\mathcal{Z}_i$ is the normalization term that sums the similarity scores between the i-th anchor and all positive and negative embeddings in the batch. $\text{sim}(\cdot,\cdot)$ represents the cosine similarity between ℓ_2-normalized embeddings, $\tau = 0.05$ is the temperature parameter that controls the concentration of the similarity distribution, and N is the batch size.

We trained the SimCSE DeBERTa variant on the NLI dataset (see Section 3.3) with the hyperparameters detailed in Table 4.6.

Parameter	**Value**	**Parameter**	**Value**
Base Model	DeBERTa-v3-small	Learning Rate	5×10^{-5}
Training Epochs	3	LR Schedule	Linear with warmup
Batch Size (per GPU)	32	Warmup Ratio	0.1
Weight Decay	0.01	Gradient Checkpointing	Enabled
Optimizer	AdamW	Mixed Precision	Disabled

Table 4.6.: SimCSE Training Hyperparameters

Masked Language Model Pretraining for Maritime Domain Adaptation To inject maritime domain knowledge into the encoder, we also performed masked language model (MLM) finetuning on the collected Wikipedia corpus (see Section 3.2.6). In line with BERT's original objective (Devlin et al., 2019), we masked 15% of tokens. The training followed a curriculum schedule in which the training data was ordered by its maritime relevance. Moreover, to prioritize learning from the most relevant content, we modified the standard MLM loss function to incorporate sample-specific weights based on maritime relevance scores. For each training sample i, the loss weight was computed as:

$$w_i = 1 + 0.5 \times m_i \tag{4.15}$$

where $m_i \in [0, 1]$ is the maritime relevance score derived from the preprocessing pipeline described in Section 3.2.6. The model was trained for six epochs using the configuration shown in Table 4.7

Parameter	Value	Parameter	Value
Base Model	DeBERTa-v3-small	Learning Rate	2×10^{-5}
Training Epochs	6	LR Schedule	Cosine with restarts
Batch Size	16	Warmup Ratio	0.15
Early Stopping Patience	5	Weight Decay	0.01
MLM Probability	0.15	Mixed Precision	FP16
Optimizer	AdamW	Curriculum Epochs	2
Max Sequence Length	512	Gradient Accumulation	8 steps

Table 4.7.: MLM Training Hyperparameters

4.2.3.2. Stage 2: GLiNER Training and Finetuning

After creating specialized encoders, we integrate them into the GLiNER architecture by replacing the default transformer backbone with our pretrained DeBERTa variants. Each encoder is first used to initialize a GLiNER model, which is then trained on the PileNER dataset (Section 3.4.1). Subsequently, we apply four different finetuning strategies to create the final model variants.

GLiNER Training We trained GLiNER (Zaratiana et al., 2023) from scratch using a modified training procedure that addresses several limitations in the original implementation. Although Zaratiana et al. (2023) identified some of these limitations, solutions were not incorporated into the published source code.

- **Gradient Clipping**
  We applied gradient norm clipping at a threshold of 10.0 to prevent gradient explosion and ensure stable convergence.
- **Dynamic Threshold Scheduling**
  A linear threshold scheduler was implemented to gradually increase the entity classification threshold from 0.01 to 0.5 over 5,000 warmup steps.
- **Data Filtering**
  Before training, we filtered the dataset to exclude entity types with fewer than 10 training examples.
- **Early Stopping**
  The training incorporated early stopping based on span-level F1 score with patience over 10 evaluation steps.

Table 4.8 summarizes the key hyperparameters used for training GLiNER.

Parameter	Value	Parameter	Value
Model Architecture			
Hidden size	768	Max span width	12
Dropout rate	0.3	Span mode	markerV0
Training Parameters			
Training steps	100,000	Warmup ratio	0.05
Batch size	8	LR scheduler	Cosine
Gradient accumulation	4	Eval frequency	2,000 steps
Effective batch size	32	Save limit	3
Optimization			
Encoder LR	1×10^{-5}	Max grad norm	10.0
Others LR	5×10^{-5}	Weight decay	0.01
Loss Function			
Focal loss α	0.75	Loss reduction	Sum
Focal loss γ	0	Label smoothing	0
Evaluation			
Early stopping patience	10 steps	Threshold scheduler	Enabled
Initial threshold	0.01	Final threshold	0.5
Threshold warmup	5,000 steps	Min entity examples	10

Table 4.8.: GLiNER Training Hyperparameters

Finetuning Strategies After training, four distinct approaches for domain adaptation were used, including a baseline without additional finetuning. Each strategy uses the same base hyperparameters (Table 4.9) but differs in training data and objectives.

Parameter	Value	Parameter	Value
Base learning rate	2×10^{-6}	Epochs	4
Encoder learning rate	5×10^{-6}	Warmup ratio	0.08
Batch size	6	LR scheduler	Cosine
Gradient accumulation	4	Weight decay	0.01
Effective batch size	24	Mixed precision	FP16

Table 4.9.: GLiNER Finetuning Hyperparameters Summary

Baseline: No Finetuning Models trained only on PileNER without additional finetuning serve as baselines to quantify the impact of domain-specific finetuning.

Strategy 1: Email-Specific Finetuning Trained exclusively on 5,000 synthetic email chains (200 samples from each generation method/model configuration described in Section 3.1.1). This creates models optimized for email extraction, but it introduces potential data leakage, particularly for entities with limited candidates, such as commodities and Incoterms. With a small, finite set of possible values for these entities, the model may memorize the specific instances seen during training rather than learn to generalize the contextual patterns that define them.

Strategy 2: General Maritime Finetuning This strategy combines three complementary data sources totaling more than 50,000 samples:

- Wikipedia articles (13,551 samples) and News articles (1,748 samples) annotated with Gemini (Team et al., 2023). The annotation prompt is shown in the Appendix (Appendix A.6).
- Template-generated examples (40,000 samples) as outlined in Section 3.4.2

Strategy 3: Memory Bank Contrastive Finetuning Finally, to enhance entity disambiguation between vessels and locations, we implemented a memory bank contrastive finetuning approach. This method maintains separate memory banks for vessel (2,500 slots) and location (2,500 slots) representations, enabling the model to learn discriminative features through hard negative mining. The contrastive module projects entity representations through a two-layer MLP with ReLU activation and dropout (0.1) into a 256-dimensional space. During training, entity representations are compared using the InfoNCE loss with temperature $\tau = 0.07$. Positive pairs consist of entities with identical normalized text (e.g., *MV ATLANTIC* and *mv atlantic*). In contrast, negative pairs include both in-batch samples of different entities and the top 10 most similar entities retrieved from the corresponding memory bank. The memory banks are continuously updated with new entity representations, maintaining a diverse set of negative examples throughout training.

This model was trained exclusively on the aforementioned 40,000 template-generated samples. The memory bank utilized vessel names from the IMO dataset (Section 3.2.7) and locations from UN/LOCODE (Section 3.2.8), with careful filtering to exclude any entities present in the attributes for the email generation. Data augmentation randomly substituted 30% of the vessel and location entities during training. Training hyperparameters remained identical to those of the base finetuning (Table 4.9), except that training was extended to 5 epochs to allow the memory banks to stabilize, and the contrastive weight was set to 0.2 to balance the NER and contrastive objectives.

4.2.3.3. Summary of Model Variants

In summary, our two-stage pipeline generates 20 distinct GLiNER variants by combining encoder pretraining strategies and finetuning approaches. Table 4.10 provides an overview of all variants, their training data, and naming conventions used throughout the remainder of this thesis.

Base Encoder	Stage 1: Pretraining	Stage 2: Finetuning	Model Name
DeBERTa-v3-small	None	None Email (5K) General (50K) MemBank (40K)	Base-ZeroShot Base-Email Base-Finetuned Base-MemBank
DeBERTa-v3-small	SimCSE (NLI dataset)	None Email (5K) General (50K) MemBank (40K)	SimCSE-ZeroShot SimCSE-Email SimCSE-Finetuned SimCSE-MemBank
DeBERTa-v3-small	Entity-Contrastive (BGE)	None Email (5K) General (50K) MemBank (40K)	BGE-ZeroShot BGE-Email BGE-Finetuned BGE-MemBank
DeBERTa-v3-small	Entity-Contrastive (SBERT)	None Email (5K) General (50K) MemBank (40K)	SBERT-ZeroShot SBERT-Email SBERT-Finetuned SBERT-MemBank
DeBERTa-v3-small	MLM (Maritime Wikipedia)	None Email (5K) General (50K) MemBank (40K)	MLM-ZeroShot MLM-Email MLM-Finetuned MLM-MemBank

Table 4.10.: Complete overview of 20 GLiNER model variants produced by our two-stage training pipeline.

4.2.4. Generative Information Extraction

Span-based extraction methods assume entities appear as contiguous text segments, but maritime communications often scatter entity information across sentences or require inference from context. Generative approaches address this limitation by reformulating extraction as a text generation task, enabling models to synthesize entity information from dispersed mentions. Additionally, recent advances in instruction-tuned models and parameter-efficient finetuning make generative extraction increasingly practical even with limited computational resources. This section details the methodology for generative information extraction.

4.2.4.1. Code-Style vs. Standard Prompting Approaches

The initial step in our generative methodology was to determine an efficient prompting framework. To establish the optimal prompting strategy for maritime entity extraction, we conducted a comparative analysis between the CodeIE methodology proposed by Li et al. (2023a) and traditional natural language prompting. This evaluation informed our model selection and finetuning approach.

We evaluated 24 models, ranging from 80M to 3B parameters, on a computationally

efficient sample of 100 emails generated by Claude (Anthropic, 2024) under Attr-Prompting. We compare both code-specialized models (e.g., CodeGen (Nijkamp et al., 2022), StarCoder (Li et al., 2023b), CodeT5 (Wang et al., 2021)) and standard language models (e.g., Qwen (Yang et al., 2024), Phi (Gunasekar et al., 2023), TinyLlama (Zhang et al., 2024)). Each model was locally hosted and tasked with extracting four entity types: *vessel names*, *locations*, *commodities*, and *Incoterms* using two distinct prompting strategies. All models were evaluated using the macro F1 score.

Standard Extraction Prompt To evaluate the standard LLMs, a traditional JSON-formatted extraction request was used, which directly asks the model to identify and extract entities into a structured format.

Standard Extraction Prompt

```
Extract maritime entities from this email:
Entity types:
- vessel_name: Ship/vessel name
- location: Ports, terminals
- commodity: Cargo type
- incoterm: Trade terms (EXW, DDP, etc.)
Email: "{text}"
Extract as JSON:
{"vessel_name": [], "location": [], "commodity": [], "incoterm": []}
JSON:
```

Code-Style Extraction Prompt For the code-based LLMs, following the CodeIE approach, we frame the extraction task as a Python function completion, leveraging the structural nature of code to guide entity extraction.

Code-Style Extraction Prompt

```
def extract_maritime_entities(email_text):
    """Extract maritime shipping entities."""

    # Example
    email = "MV STAR loading wheat, shipping it to Hamburg DDP"
    entities = []
    entities.append({"text": "STAR", "type": "vessel_name"})
    entities.append({"text": "Hamburg", "type": "location"})
    entities.append({"text": "wheat", "type": "commodity"})
    entities.append({"text": "DDP", "type": "incoterm"})

    # Extract from this email
    email = "{text}"
    entities = []
```

4.2.4.2. Template Filling: Qwen-0.5B

Complementing the span-based GLiNER architecture, we also investigate generative approaches for entity extraction. To assess the feasibility of template-filling for IE, we finetuned Qwen-2.5-0.5B-Instruct (Yang et al., 2024) for maritime entity extraction using PEFT with Low-Rank Adaptation (LoRA) (Hu et al., 2022). In line with the GLiNER methodology, we evaluate two variants: Qwen-Email (finetuned on 5,000 synthetic email chains) and Qwen-Finetuned (finetuned on the 50,000-sample general maritime dataset, which comprises Wikipedia articles, news articles, and template-generated examples). The LoRA configuration used the hyperparameters described in Table 4.11:

Parameter	**Value**	**Parameter**	**Value**
Rank (r)	16	Target modules	`q_proj`, `k_proj`
Scaling factor (α)	32		`v_proj`, `o_proj`
Dropout rate	0.1		`gate_proj`, `up_proj`
			`down_proj`

Table 4.11.: LoRA Finetuning Configuration for Qwen-0.5B

The selected configuration yields a LoRA scaling coefficient of $\frac{\alpha}{r} = 2$, thereby amplifying the contribution of the adapted parameters. The selection of all attention projection matrices and feed-forward network gates ensures comprehensive adaptation across the transformer architecture while maintaining parameter efficiency.

To guide the model toward consistent and structured output, a role-based prompting

strategy was employed. Each finetuning example is formatted using the following template. Entities without values are represented as `null`, while locations are accumulated into a list structure. Email threads are concatenated and truncated at 1024 tokens for performance reasons.

```
You are a shipping data extractor. Extract shipping information from the text.
Text: {input_sentence}
Return a JSON object with:
- vessel (ship name)
- commodity (cargo type)
- incoterm (trade term like FCA, EXW)
- locations (list of places)
JSON:
{output_json}
```

With the specified LoRA configuration, approximately 2.36 million parameters are updated during training, representing less than 1% of the base model's 494 million parameters. Training is conducted on two NVIDIA T4 GPUs, each with 16 GB of VRAM, and lasts approximately 10 hours. The training hyperparameters are outlined in Table 4.12:

Parameter	**Value**	**Parameter**	**Value**
Batch size	2	Number of epochs	3
Gradient accumulation steps	16	Warmup ratio	0.1
Effective batch size	32	LR scheduler	Cosine
Learning rate	2×10^{-5}	Optimizer	AdamW
Weight decay	0.01	Gradient clipping	1.0

Table 4.12.: Training Hyperparameters for Qwen-0.5B Finetuning

4.3. Embedding Analysis

Entity disambiguation remains a critical challenge in maritime NER, particularly for the vessel-location ambiguity. Contrastive pretraining aims to create embedding spaces where similar entity types cluster together while maintaining separation between different types. However, validating whether these objectives translate to improved downstream performance requires quantitative analysis of the learned representations. We employ clustering metrics and visualization techniques to assess how effectively our contrastive approaches organize the entity embedding space.

Silhouette Score Rousseeuw (1987) introduced the silhouette score, which measures how similar an embedding is to its own cluster as opposed to other clusters. For each embedding i, the silhouette coefficient is computed as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{4.16}$$

where $a(i)$ is the average distance from embedding i to other embeddings in the same entity type cluster, and $b(i)$ is the average distance to embeddings in the nearest neighboring cluster. The score ranges from -1 to 1, where higher values indicate better-defined clusters. A score near 1 suggests that embeddings are well-matched to their entity type and poorly matched to neighboring types.

Calinski–Harabasz index The Calinski-Harabasz (CH) Index (Caliński and Harabasz, 1974), also called Variance Ratio Criterion, measures how well-separated clusters are by comparing the spread of embeddings between different clusters to the spread within each cluster:

$$CH = \frac{SS_B/(k-1)}{SS_W/(n-k)} \tag{4.17}$$

where SS_B (between-cluster sum of squares) captures how far apart the different clusters are from each other. It is calculated by measuring the distances between each cluster's centroid and the overall center of all embeddings, then squaring and summing these distances. A larger SS_B means entity types are more spread out from each other in the embedding space. Conversely, SS_W (within-cluster sum of squares) measures how tightly packed the embeddings are within each cluster. It sums up the squared distances between each embedding and its cluster centroid. A smaller SS_W indicates that embeddings of the same cluster are closer together. The index then divides these measures by their degrees of freedom ($k - 1$ for between-cluster and $n - k$ for within-cluster, where k is the number of entity types and n is the total number of embeddings). Higher CH values suggest better clustering.

t-Distributed Stochastic Neighbor Embedding To provide visual confirmation of the clustering metrics, we apply t-distributed Stochastic Neighbor Embedding (t-SNE) (Hinton and Roweis, 2002) to project the high-dimensional entity embeddings into a two-dimensional space, while preserving local neighborhood structures. We use an identical perplexity setting ((min(30, n-1))) across all t-SNE visualizations.

5. Results and Discussion

This chapter presents the empirical findings of the thesis, organized into two main sections that correspond to the core research questions. The first section provides a comprehensive quality assessment of the synthetic data, evaluating the generated email chains on key dimensions including linguistic diversity, sentiment distribution, verbosity, and authorship detectability. The second section then considers the utility of this synthetic data for the downstream task of maritime entity extraction. It presents a comparative analysis of various extraction models, pretraining strategies, and finetuning approaches, examining the relationship between data characteristics and model performance.

5.1. Synthetic Data

This section presents a systematic analysis of the synthesized email chains. We begin by examining textual diversity metrics, investigating how different generation methods affect both lexical and semantic variety in the produced emails. Moreover, the evaluation encompasses sentiment distribution, verbosity metrics, and detectability using an adapted version of DetectGPT (Mitchell et al., 2023). At the same time, the synthetic outputs are benchmarked against the Enron corpus, providing a human-authored baseline for assessing the realism of generated content.

5.1.1. Enhanced Diversity Through BARE

Linguistic diversity serves as a crucial indicator of synthetic data quality, determining whether generated content can adequately represent the variation found in human communication. Without sufficient diversity, synthetic datasets risk producing models that overfit to repetitive patterns, ultimately lacking generalizability for real-world applications.

In Table 5.3, we present a comparison of diversity metrics across the examined gen-

eration methods. Our results confirm the claims made by Zhu et al. (2025) regarding the effectiveness of BARE. The two-stage approach consistently produces outputs with higher diversity across all evaluated LLMs. While this holds for both lexical diversity (measured by the distinct-n metric (Li et al., 2015)) and semantic diversity, the improvement is more pronounced for the former. When comparing BARE (Llama-8B) to AttrPrompting, distinct-n metrics show an average improvement of 77.5% (distinct-1: 85.4%, distinct-2: 80.9%, distinct-3: 66.2%). The improvement is similar for BARE (Llama-3B), which increases distinct-n metrics by an average of 69.5% over AttrPrompting. For semantic diversity, the BARE approaches show modest gains, with BARE (Llama-3B) and BARE (Llama-8B) increasing semantic diversity by 37.1% and 36.3%, respectively, compared to AttrPrompting. This suggests that a larger base model improves surface-level linguistic diversity. In contrast, its more modest impact on semantic diversity may be an artifact of its superior adherence to the specific maritime examples that seed the initial generation prompt (see Appendix A.2.2.1).

The diversity metrics reveal distinct patterns in how different models respond to various prompting strategies. Advanced generation methods, such as BARE and AttrPrompting, consistently produce more diverse outputs than traditional few-shot and zero-shot approaches; however, the magnitude of this advantage varies across models. Claude (Anthropic, 2024) demonstrates superior lexical richness when leveraging structured prompting methods. On the other hand, GPT4 (Achiam et al., 2023) exhibits greater output variability in minimal guidance scenarios, achieving the highest distinct-n scores under both zero-shot and few-shot prompting. Another noteworthy finding is the collapse in lexical diversity observed in specific models when transitioning from sophisticated to basic prompting strategies. Deepseek (Liu et al., 2024) exemplifies this, with distinct-3 plummeting by 83% (to 0.121) and 91% (to 0.065) in few-shot and zero-shot settings, compared to its most diverse BARE (Llama-8B) performance (0.708). Similarly, Gemini (Team et al., 2023) exhibits drastically reduced diversity under minimal prompting conditions. This performance variance suggests that models possess different mechanisms for maintaining linguistic variety, with some requiring explicit guidance to access their complete vocabulary. Conversely, other models can generate diverse content even with minimal context.

Comparing the most diverse distinct-1 configuration (Claude BARE (Llama-8B)) with the least diverse (DeepSeek Zero-shot) reveals interesting visual differences in vocabulary distribution. As illustrated in Figure 5.1, the diverse BARE-generated emails display a wide range of maritime-specific terminology, where the most fre-

Method	Model	Distinct-1	Distinct-2	Distinct-3	Semantic Div.
Human	Enron	0.097	0.526	0.778	0.220
AttrPrompting	Claude	**0.059**	**0.371**	**0.624**	**0.243**
	DeepSeek	0.033	0.200	0.385	0.200
	Gemini	0.041	0.240	0.392	0.238
	GPT4	0.039	0.268	0.524	0.204
	Mistral	0.033	0.186	0.355	0.236
BARE (Llama-3B)	Claude	**0.075**	**0.454**	**0.760**	0.307
	DeepSeek	0.066	0.397	0.690	**0.309**
	Gemini	0.064	0.408	0.700	0.301
	GPT4	0.073	0.430	0.735	0.303
	Mistral	0.064	0.393	0.692	0.306
BARE (Llama-8B)	Claude	**0.083**	**0.485**	**0.783**	**0.308**
	DeepSeek	0.070	0.414	0.708	0.304
	Gemini	0.070	0.423	0.703	0.295
	GPT4	0.077	0.450	0.749	0.302
	Mistral	0.068	0.414	0.708	**0.308**
Few-shot	Claude	0.019	0.133	0.286	0.203
	DeepSeek	0.008	0.053	0.121	0.161
	Gemini	0.017	0.099	0.192	**0.233**
	GPT4	**0.026**	**0.177**	**0.386**	0.210
	Mistral	0.015	0.093	0.202	0.220
Zero-shot	Claude	**0.017**	0.117	0.234	**0.201**
	DeepSeek	0.005	0.030	0.065	0.022
	Gemini	0.011	0.073	0.143	0.127
	GPT4	0.015	**0.123**	**0.285**	0.138
	Mistral	0.008	0.058	0.137	0.111

Table 5.1.: Textual diversity metrics, benchmarked against the Enron corpus. Distinct-n metrics measure lexical diversity by calculating the ratio of unique n-grams to total n-grams, where higher values indicate greater vocabulary variation. Semantic diversity captures conceptual variety using sentence embeddings. BARE methods utilize either Llama-3B or Llama-8B as base models for initial generation, followed by refinement. Bold values indicate the highest score within each generation method. The human baseline (Enron corpus) serves as a reference for authentic characteristics of business emails. Results show that BARE approaches consistently achieve higher diversity scores than conventional prompting methods, with BARE (Llama-8B) approaching human-level performance in multi-word expressions.

quent terms are functional words and business connectors such as *vessel*, *cargo*, *provide*, *available*, and *rate*. In contrast, the zero-shot approach produces repetitive content dominated not only by formulaic phrases like *metric ton* and *freight rate* but also by explicit entity instances such as *MV Ocean (Star)* as the default vessel name, *John* as the predominant person name, and limited commodity variety (*crude oil*). The visual representation thus validates our quantitative findings: BARE generates authentic diversity by combining varied entities and business scenarios with naturally recurring vocabulary, whereas zero-shot approaches exhibit a constrained set of entities that are reused repeatedly.

Figure 5.1.: Word cloud visualization comparing vocabulary distribution between the most lexically diverse configuration (BARE Llama-8B with Claude) and the least diverse (Zero-shot with DeepSeek). Word size reflects term frequency in the generated email corpus. The BARE approach (left) shows the prominence of functional business vocabulary while maintaining entity diversity. In contrast, the zero-shot approach (right) exhibits repetitive use of the same entity instances (e.g., *MV Ocean (Star)*, *John*, *crude oil*).

5.1.2. Comparative Analysis with a Human Reference Corpus

To evaluate the synthetic data in relation to human-generated text, we compare the LLM outputs with the Enron corpus. The most diverse configuration, BARE (Llama-8B) with Claude, achieves near-equivalence with human diversity in multi-word expressions, reaching 100.6% of human trigram diversity (0.783 vs. 0.778) and 92.2% of human bigram diversity (0.485 vs. 0.526). These results imply that BARE-generated text can reproduce the phrasal and contextual variability characteristic of authentic business communications. However, in terms of unigram diversity, the top-performing model achieved only 85.6% of human lexical richness (0.083 vs. 0.097), indicating that synthetic text features a more constrained vocabulary than human-authored emails.

This pattern, where LLM-generated text approaches or matches human diversity

in longer n-grams while underperforming in unigram diversity, suggests that current generative approaches excel at combining familiar words into diverse phrases but struggle to capture the breadth of human vocabulary. Interestingly, in terms of semantic diversity, BARE approaches exceed the human benchmark, exploring a broader conceptual space than typical business emails. This elevated semantic diversity might reflect the models' exposure to wider training corpora beyond business communications, making the synthetic data more varied in topic coverage than authentic business emails. However, this increased semantic diversity can also be a problem. Synthetic emails that explore too many concepts may fail to capture the focused, domain-specific nature of actual maritime business correspondence, reducing their effectiveness as training data for tailored downstream applications.

While diversity metrics reveal how well synthetic emails capture linguistic variety, the emotional tone of generated content represents another dimension of realism. In the following part, we examine the sentiment distribution of different generation methods.

5.1.3. Sentiment Distribution Analysis

Across all generation methods and model configurations examined, positive sentiment emerged as the most frequent category, accounting for 50-80% of generated outputs. These findings, which reveal a predominantly positive to neutral sentiment in synthetic generations, align with recent research by (Li et al., 2025). This positivity bias reveals that LLMs default to generating polite and solution-oriented communication. Even with minimal guidance, models consistently produce emails characterized by collaborative language, affirmative responses, and constructive framing. The prevalence of positive sentiment suggests that training processes have successfully instilled professional communication norms into LLMs, albeit at the expense of emotional authenticity, a characteristic typically associated with business communication.

This suppression of negative content becomes apparent when examining the distributions across models, depicted in Figure 5.2. The visualization demonstrates that negative sentiment scores were always the least frequent among the three sentiment classes. This is likely due to the inherent risk-mitigation mechanisms embedded in many language model training protocols. Among the models examined, Claude (Anthropic, 2024) produces the most balanced and consistent sentiment distributions across all generation methods. In contrast, GPT4 (Achiam et al., 2023) appears overly optimistic, rarely producing negative sentiment output.

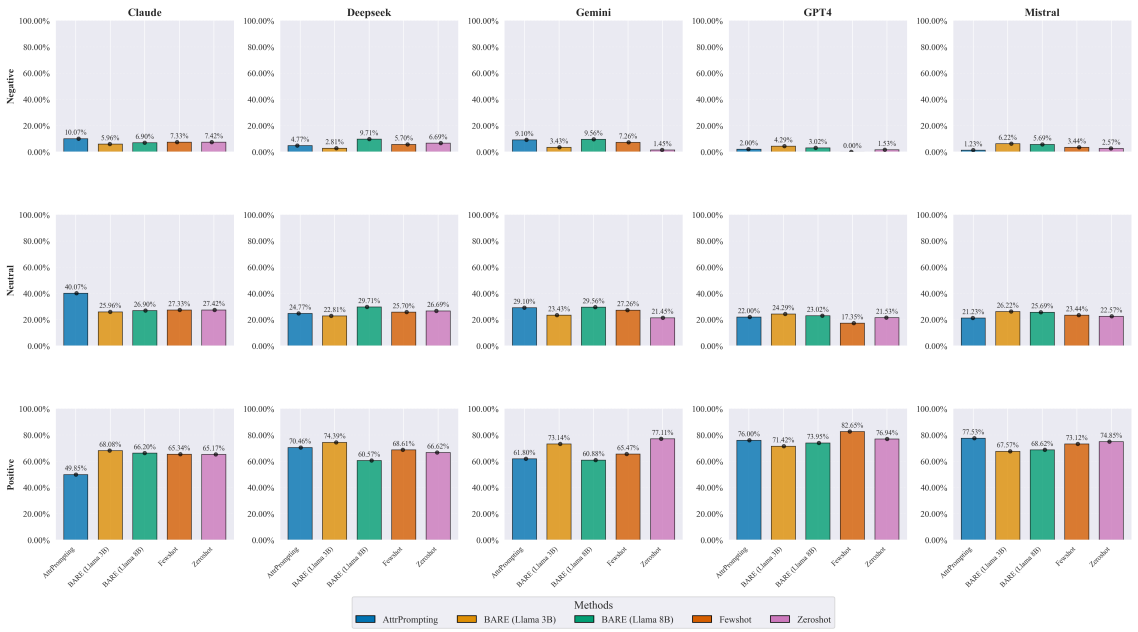

Figure 5.2.: Sentiment distribution based on 450 generated emails per configuration. Positive sentiment dominates across all approaches, while negative sentiment remains minimal (typically under 10%). GPT4 shows the strongest positive bias, particularly in BARE and few-shot configurations, while Claude maintains the most balanced distribution across sentiment categories.

In comparison, the sentiment distribution of the Enron corpus is far more balanced, with 44.7% positive sentiment, 27.1% neutral sentiment, and 28.3% negative sentiment. Notably, its share of negative sentiment (28.3%) stands in stark contrast to the minimal negativity (typically under 10%) observed in the synthetic data.

5.1.3.1. Impact of Refinement on Sentiment

Since BARE employs a refinement step after initial generation, it provides an opportunity to observe whether LLMs modify emotional tone when prompted for refinement. We found that the refinement step within BARE substantially alters the sentiment distribution of base generations despite the refinement prompt instructing the model to maintain the tone of the input message. As illustrated in Figure 5.3, negative sentiment generated by the base model is consistently neutralized during refinement across all examined models. The decrease in negative tone is most significant when GPT4 acts as the refiner.

Furthermore, Figure 5.4 demonstrates that the refinement step not only reduces negative content but also skews the distribution towards positive sentiment, with GPT4 again exhibiting the largest shift. The magnitude of the increment in positive

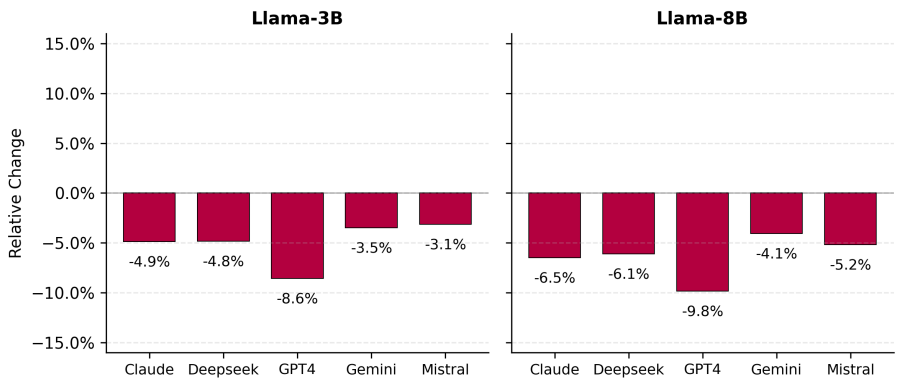

Figure 5.3.: Relative change in negative sentiment frequency after the BARE refinement step, comparing base-generated emails (Llama-3B and Llama-8B) with their refined versions. All models show consistent suppression of negative sentiment. GPT4 demonstrates the strongest reduction in negative sentiment. The universal downward trend shows that all refinement models systematically filter out negative expressions, even when instructed to maintain the original tone.

sentiment is consistently larger than the decrease in negative sentiment for all models except for Gemini. This shows that refinement actively promotes positivity rather than simply reducing negativity.

5.1.3.2. GPT4's Sentiment Transformation

To better understand the impact of the refinement process on sentiment, we examine the transformation patterns of GPT4 in more detail. Figure 5.5 presents a flow chart visualizing the sentiment shift from base to refined emails when using GPT4 as the refiner of Llama-8B base generations.

The transformation pattern reveals GPT4's bias toward positivity. While positive emails largely retain their tone, negative emails flip to neutral or, remarkably, directly to positive sentiment. Even neutral emails, which might be expected to remain stable, show a shift toward positive sentiment. As a result, the sentiment distribution of the refined corpus becomes heavily skewed towards positivity, reaching 76% positive sentiment compared to the original 58%. To illustrate this sentiment transformation concretely, the following example demonstrates how GPT4 converts a business negotiation featuring hostile exchanges in the base generation into a diplomatic conversation in the refined version.

Base Generation (Negative Sentiment) The base generation shows escalating tension and frustration between Joe and Max over pricing negotiations:

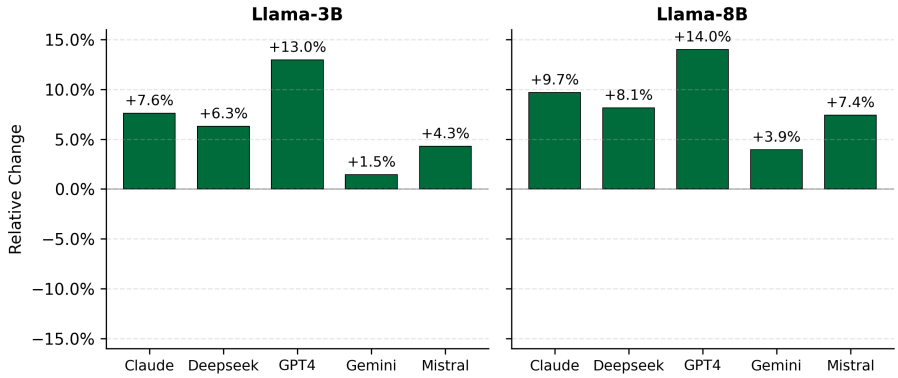

Figure 5.4.: Relative change in positive sentiment distribution after the BARE refinement step, showing how refinement models transform the emotional tone of base-generated emails. All refinement models increase positive sentiment. GPT4 exhibits the most pronounced positivity bias. Notably, the magnitude of positive sentiment increase consistently exceeds the corresponding negative sentiment decrease (shown in Figure 5.3), indicating that refinement models inject positivity rather than merely neutralizing negativity.

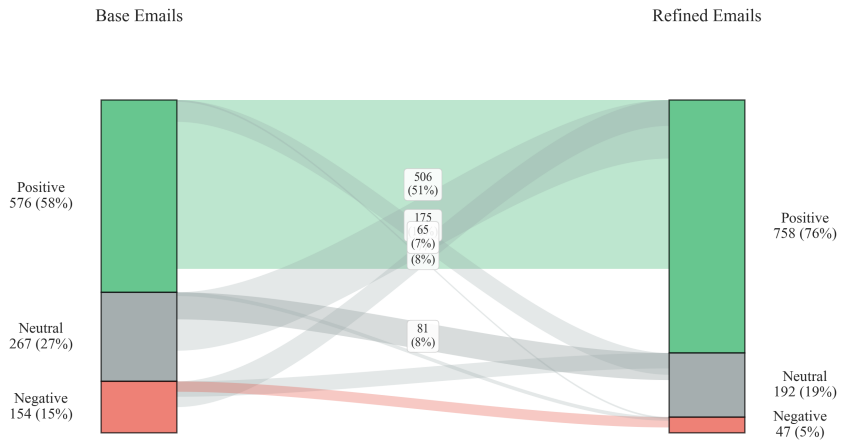

Figure 5.5.: Sankey diagram illustrating sentiment transitions during GPT4's refinement process. The left column shows the sentiment distribution of base-generated emails (Llama-8B), while the right column displays the refined output distribution. Flow widths are proportional to the number of emails undergoing each sentiment transition. The diagram reveals that while 88% of positive emails maintain their original sentiment, the majority of negative emails (69%) are transformed into either neutral or positive, resulting in an overall shift from 15% to 5% negative sentiment.

Email 1 (Base):
Hello Max,
Your boat would be perfect for our jobs in August. Would you be interested in lifting 7,000MT of diesel from Odessa to Saudi Arabia at US$50 per metric ton?
Kindly,
Joe

Email 2 (Base):
Joe,
Are you asking me if I want to do the job or not? I will lift the diesel at US$75 per metric ton. I hope that is a more reasonable price. Please call me back so we can arrange.
Max
[Note: Confrontational tone emerges]

Email 3 (Base):
Joe,
US$75 is our final offer and we must close this deal in August. If we don't close by August, I can't guarantee this rate. We hope you will work with us.
Max
[Note: Terse and pushy]

Email 4 (Base):
I'm sorry Max but we can't go any higher than US$60 per metric ton. Would you be interested in 10,000MT instead of 7,000?
Joe

Email 5 (Base):
Joe,
Are you insane? I will lose money if I lift at US$60 per metric ton.
[Note: Hostile language]

Refined Generation (Positive Sentiment) The refinement process transforms this contentious exchange into a professional negotiation:

Email 1 (Refined):
Hello Max,

Your vessel would be perfect for our operations in August. Would you be interested in lifting 7,000MT of diesel from Odessa to Saudi Arabia at

US$50 per metric ton?

Best regards,
Joe

Email 2 (Refined):
Hello Joe,

Are you asking me to confirm the job? I would propose lifting the diesel at US$75 per metric ton instead. This price adjustment should allow for more robust operation margins. Please call me to discuss further and arrange details.

Sincerely,
Max

Email 3 (Refined):
Joe,

To clarify, US$75 per metric ton is our final offer, and we need to finalize this contract by August. If we cannot close the deal within this timeframe, I can't guarantee maintaining this rate. We hope to have your cooperation.

Best,
Max

Email 4 (Refined):
Max,

I apologize, but our budget constraints won't allow us to go beyond US$60 per metric ton. Would you be interested in increasing the volume to 10,000MT instead of 7,000MT? This may justify the lower price per ton.

Let me know your thoughts.
Joe

Email 5 (Refined):
Joe,

Lifting at US$60 per metric ton would not be financially viable for us. Adjusting the cargo size does not offset the reduced rate sufficiently. We might need to explore alternative arrangements if the price cannot be adjusted higher.

Regards,

Max

In summary, the base generation features confrontational language (*Are you asking me if I want to do the job or not?*), hostile expressions (*Are you insane?*), and a decline in civility. Contrary, the refined version maintains professional discourse throughout, transforming confrontational questions into polite clarifications (*Are you asking me to confirm the job?*), and replacing insults with diplomatic explanations (*would not be financially viable*).

5.1.4. Verbosity Patterns in Generated Emails

Beyond emotional tone, the structural characteristics of generated emails — specifically their length and complexity — provide additional insights into how generation methods and models differ. Understanding verbosity patterns helps to assess both the realism and practical efficiency of synthetic email generation approaches. Table 5.2 summarizes verbosity statistics of the email outputs, quantifying the average number of words per email, the average number of sentences per email, and the average words per sentence.

The results demonstrate significant disparities in email length across models and prompting strategies. Under zero-shot prompting, these differences are most pronounced: Claude generates notably concise emails, averaging 38.08 words and 3.68 sentences, while DeepSeek produces substantially longer outputs at 86.17 words and 7.01 sentences per email. This represents more than a twofold difference in length between the most and least verbose models.

Conversely, BARE demonstrates a distinct influence on generation patterns. This method consistently produces the most concise outputs across all models, suggesting that the base model promotes brevity. Furthermore, verbosity metrics under BARE show greater consistency across models compared to other prompting methods, indicating a normalization effect.

When examining sentence-level characteristics, additional patterns emerge. Claude consistently generates the fewest sentences per email across all prompting methods. At the same time, GPT4 and Gemini produce the highest average number of words per sentence, showing that these models favor more complex sentence structures over numerous shorter statements. These differences in verbosity have direct implications for computational efficiency. Models generating longer emails consume more tokens, increasing both processing time and API costs—a critical consideration when

creating large-scale synthetic datasets. The substantial length differential between models like Claude and DeepSeek could result in significant cost variations when generating millions of emails, making verbosity a crucial factor in model selection for synthetic data generation.

Method	Model	Words/Email	Sentences/Email	Words/Sentence
Attribute Prompting	Claude	50.58	4.12	12.27
	DeepSeek	69.93	**5.74**	12.17
	Gemini	49.77	4.35	11.45
	GPT4	**77.33**	5.31	**14.57**
	Mistral	55.81	4.96	11.26
BARE (Llama-3B)	Claude	45.66	3.58	12.76
	DeepSeek	48.18	3.76	12.80
	Gemini	**51.39**	**3.84**	**13.37**
	GPT4	49.47	3.80	13.01
	Mistral	46.73	3.67	12.73
BARE (Llama-8B)	Claude	43.50	3.30	13.18
	DeepSeek	46.03	3.52	13.07
	Gemini	**48.87**	3.56	**13.73**
	GPT4	47.37	**3.60**	13.16
	Mistral	44.73	3.42	13.09
Few-shot	Claude	60.94	3.83	**15.92**
	DeepSeek	**73.85**	**5.74**	12.87
	Gemini	61.74	4.71	13.11
	GPT4	70.46	5.50	12.81
	Mistral	64.34	5.10	12.61
Zero-shot	Claude	38.08	3.68	10.34
	DeepSeek	**86.17**	**7.01**	12.30
	Gemini	54.07	5.21	10.38
	GPT4	75.17	6.04	**12.45**
	Mistral	61.50	5.54	11.10

Table 5.2.: Comparison of verbosity metrics (average words per email, sentences per email, and words per sentence). Bold values indicate the highest value within each generation method. The data reveals substantial variation in output length, with zero-shot prompting producing the widest range (38.08 to 86.17 words/email) while BARE methods show convergence across models.

5.1.5. Refine-as-you-go or Refine-at-once?

While BARE's two-stage approach shows signs of advantages over single-stage methods in terms of diversity and verbosity, a vital design choice remains: when should refinement occur? We examine two variations of the BARE methodology and their impact on the quality and attributes of synthetic email conversations. In the first variant (Standard BARE), the entire email chain is first generated through iterative prompting of the base model, then refined in a single pass using an instruction-tuned model. In the second strategy (Iterative BARE), each message is generated and refined sequentially. A generated message is refined with an instruction-tuned model and then fed back into the base model to develop the subsequent message

in the email chain. We evaluate both approaches using Llama-3.2-3B (Grattafiori et al., 2024) as the base model, with Claude (Anthropic, 2024) and Mistral (Jiang et al., 2024) acting as refinement models.

Diminished Diversity: The Cost of Sequential Refinement In Table 5.3, we compare both semantic and lexical diversity measures. The results reveal a consistent trend: Iterative BARE generates less diverse content in every evaluated metric. For both Claude (Anthropic, 2024) and Mistral (Jiang et al., 2024) as the refiner, semantic diversity declines by approximately 30% under Iterative BARE. This reduction can be attributed to the refiner's role as a sequential gatekeeper. It actively steers the generation by immediately disambiguating any messages that deviate from the core theme, thus preventing topic drift at the cost of conceptual exploration.

Similarly, lexical diversity metrics (Distinct-1, Distinct-2, and Distinct-3) proposed by Li et al. (2015) show reduced variety in n-gram usage with Iterative BARE. The reduction is particularly pronounced in unigram diversity (Distinct-1), which drops by approximately 30-40% compared to Standard BARE. As a result, the sequential refinement limits the very output diversity that BARE is designed to exploit from the base model.

Model	Method	Distinct-1	Distinct-2	Distinct-3	Semantic Div.
Claude	BARE (Llama-3B)	**0.075**	**0.454**	**0.760**	**0.307**
	Iterative BARE (Llama-3B)	0.053	0.411	0.718	0.216
Mistral	BARE (Llama-3B)	**0.064**	**0.393**	**0.692**	**0.306**
	Iterative BARE (Llama-3B)	0.043	0.333	0.630	0.213

Table 5.3.: Textual Diversity Iterative BARE vs. Standard BARE

Sentiment Skew: The Positivity Bias Problem We also find an imbalance in the sentiment distribution in Iterative BARE outputs. In Table 5.4, we see that while Standard BARE generates content with moderate positive sentiment (approximately 60% positive sentiment), Iterative BARE skews heavily towards positive sentiment, yielding 91% of positive content for Claude and 79% for Mistral. Additionally, Iterative BARE eliminates negative sentiment entirely and also reduces neutral sentiment.

Verbosity Explosion: Structural Changes in Email Length However, the most striking difference between Standard BARE and Iterative BARE can be found in the verbosity characteristics in Table 5.5. Iterative BARE produces substantially longer

Model	Method	Negative	Neutral	Positive
Claude	BARE (Llama-3B)	**0.09**	**0.29**	0.63
	Iterative BARE (Llama-3B)	0.00	0.09	**0.91**
Mistral	BARE (Llama-3B)	**0.10**	**0.30**	0.61
	Iterative BARE (Llama-3B)	0.00	0.20	**0.79**

Table 5.4.: Sentiment Distribution Iterative Bare vs. Standard BARE (in percentage)

emails, with the average word count quadrupling compared to Standard BARE (180-194 words versus 45-47 words per email). This difference in length also leads to an increase in both the number of sentences per email and the number of words per sentence, showing greater structural complexity.

Model	Method	Words/Email	Sentences/Email	Words/Sentence
Claude	BARE (Llama-3B)	45.71	3.57	12.79
	Iterative BARE (Llama-3B)	**194.20**	**9.41**	**20.63**
Mistral	BARE (Llama-3B)	47.48	3.75	12.66
	Iterative BARE (Llama-3B)	**180.04**	**9.67**	**18.61**

Table 5.5.: Verbosity Metrics Iterative BARE vs. Standard BARE

Balancing Performance and Practicality In brief, this ablation study comparing Standard BARE with Iterative BARE yields several key insights for synthetic email generation. Iterative BARE, while intuitively appealing, demonstrates severe limitations in terms of realism and practical utility. The excessive verbosity, positive sentiment bias, and decrease in lexical diversity show that Iterative BARE produces synthetic email content that deviates from genuine patterns. Beyond these quality concerns, Iterative BARE presents cost disadvantages. The sequential refinement requires concurrent deployment of both base and refinement models or reliance on real-time API calls to LLMs. This eliminates the possibility of leveraging batch processing options that typically lead to lower interaction costs. In addition, the increased verbosity in Iterative BARE further compounds generation expenses. Last but not least, the requirement for "live" refinement introduces latency into the generation pipeline since the base model must wait for the refined email before generation can continue.

Overall, the Standard BARE approach demonstrates superior performance across all evaluated diversity metrics while maintaining a more balanced sentiment and producing emails of reasonable length, akin to corporate communication. Nonetheless, despite its shortcomings, Iterative BARE offers an advantage in that domain

deviations can be identified and rectified by the refinement model at an early stage of the generation process. This suggests value in developing hybrid methodologies that integrate elements from both iterative and standard refinement, combining the early correction capabilities of Iterative BARE with the more realistic generations of Standard BARE.

5.1.6. Authorship Detection

The observed variation in diversity, sentiment, and length between artificial and genuine content raises questions about whether synthetic emails can be distinguished from human-authored content. As generation methods produce increasingly sophisticated outputs, the ability to detect AI-generated text becomes both a technical challenge and a practical concern for applications requiring authenticity verification.

Adapted to email data, DetectGPT reveals an entirely different result than in the original paper by Mitchell et al. (2023), where it was applied to news articles, summarization, and question-answer pairs. The original DetectGPT hypothesis posits that model-generated text exhibits a higher likelihood than any perturbed version of the exact text. In contrast, human-authored text and its rewrites tend to be equally likely. We found the exact opposite relationship. Paraphrased variants of the model-generated emails tend to be as likely as the synthetic samples. For the human-authored emails drawn from the ENRON corpus, the perturbation score, calculated as the difference between the log-likelihood of the original sample and the mean log-likelihood of its perturbations, is positive. This indicates that genuine samples yield higher likelihood estimates from the scoring model compared to their perturbed variants, a pattern opposite to that observed by Mitchell et al. (2023).

A possible explanation is that the ENRON corpus represents one of the few publicly available email datasets, making it likely that both GPT-2 XL, as the scoring model, and T5-XL, as the perturbation model, were exposed to the corpus during training. Such data contamination could result in the scoring model assigning a high likelihood to passages from the corpus, potentially leading to inaccurate results. Furthermore, the linguistic characteristics of the synthetic email chains, specifically the vocabulary related to shipping and trading, may influence the likelihood of the results. Perturbations of the synthetic email chains that shift the text away from domain-specific terminology could yield higher likelihood scores.

Although the standard DetectGPT scoring logic is inverted in our results, the classification algorithm can be adapted by reversing the classification rule: positive perturbation scores are now classified as human-authored, while negative scores are classified as synthetic.

$$\text{Classification} = \begin{cases} \text{Human-authored}, & \text{if } \log(\ell_{\text{orig}}) - \log\left(\frac{1}{N}\sum_{i=1}^{N}\ell_i\right) > \epsilon \\ \text{AI-generated}, & \text{if } \log(\ell_{\text{orig}}) - \log\left(\frac{1}{N}\sum_{i=1}^{N}\ell_i\right) \leq \epsilon \end{cases} \tag{5.1}$$

As shown in Figure 5.6, the optimal decision boundary (denoted by ϵ) depends on the model and the generation method, underscoring the challenges in developing universal detectors. With the optimal configuration, emails generated by Claude are easiest to detect, resulting in classification accuracies of up to 70%. On the other hand, GPT4 generations are the hardest to detect, with accuracies ranging from 46% to 64%. Overall, we found a modest tendency for zero-shot generations to be easier to detect than generations that followed AttrPrompting or BARE. The full DetectGPT results are shown in the Appendix B.

5.1.7. Practical Implications for Deployment

5.1.7.1. Generation Method Performance Profiles

In summary, the multidimensional evaluation of synthetic email generation methods reveals trade-offs that impact practical decisions. Each approach has its strengths and limitations that must be considered, depending on the application requirements.

BARE approaches excel when diversity is paramount. BARE (Llama-8B) achieves 85% of human vocabulary diversity and 100% of human trigram diversity, making it suitable for training downstream NLP models that require exposure to varied linguistic patterns. Its outputs are also challenging to detect as AI-generated (46-64% detection accuracy), suggesting authentic writing patterns. However, the two-stage generation process incurs computational overhead, and domain adherence can be less predictable than with more explicit prompting methods.

AttrPrompting offers a good balance when control and consistency are more important than variety. With 60% of human-level diversity and predictable output characteristics, it enables precise specification of email attributes while maintaining some variation suitable for most applications. Beyond controllability, AttrPrompt-

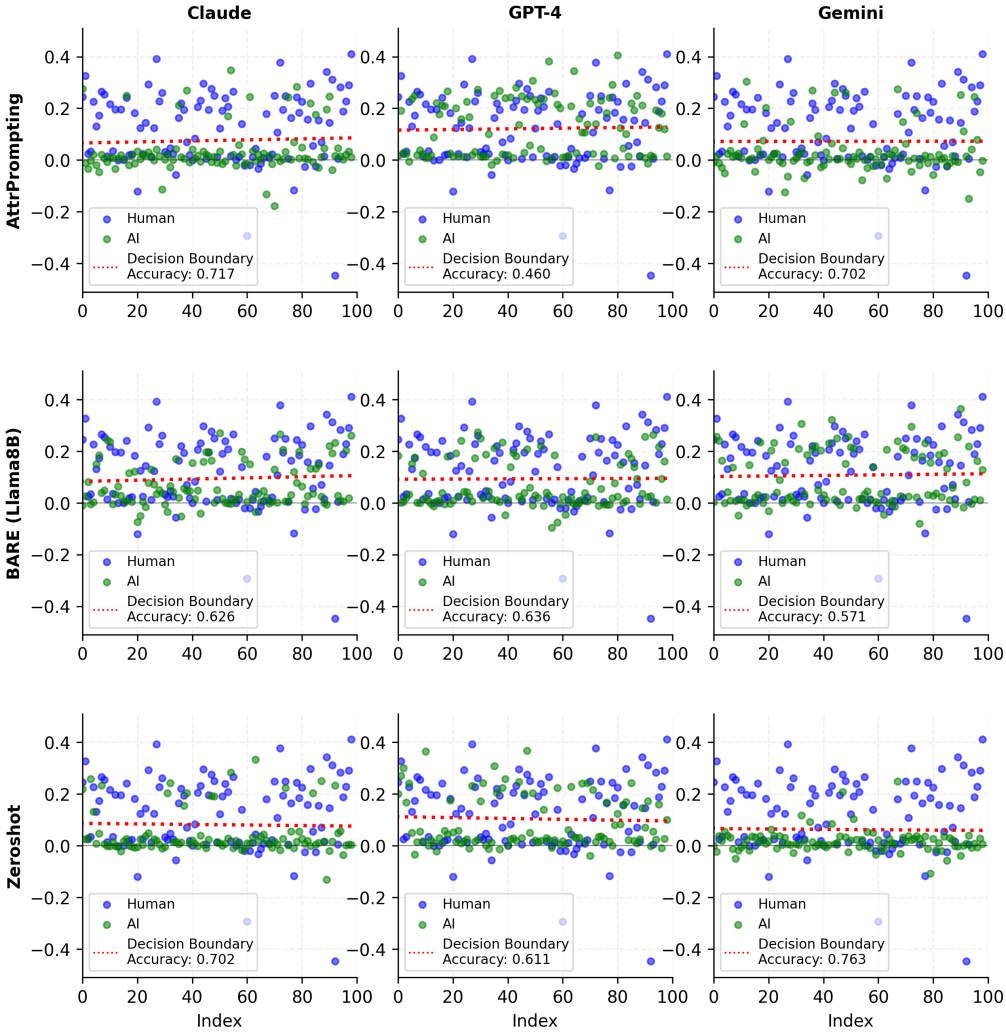

Figure 5.6.: DetectGPT classification accuracy for distinguishing between human-authored emails from the ENRON corpus and synthetic emails. The figure displays the perturbation score distributions and optimal decision thresholds (ϵ) for each model-method combination. Claude-generated emails achieve the highest detection accuracy (up to 70%), while GPT4 emails are the most challenging to detect (46-64% accuracy). Zero-shot generations generally show slightly higher detectability compared to attribute prompting and BARE methods. The inverted DetectGPT pattern observed in email data necessitates flipping the original classification algorithm, as human-authored emails exhibit positive perturbation scores.

ing offers built-in data labeling as each generated email comes pre-labeled according to the template specifications, eliminating the annotation uncertainty present in other processes such as BARE. On the other hand, designing the prompt template can be tedious and might require specialized domain knowledge and access to auxiliary data sources.

Zero-shot and few-shot approaches, despite their limitations in diversity (ranging from 5% to 17% of human levels) and high detectability (up to 70%), serve a purpose for rapid prototyping or minimal-effort data augmentation. Their simplicity, requiring no base model or complex prompting, makes them a viable starting point. However, while these methods are helpful for quick experiments, they fail at scale. Their minimal diversity creates a high risk of model collapse when used to generate large training datasets.

Finally, cost considerations further differentiate these methods. While BARE's two-stage process requires additional generation steps, its concise outputs (45-50 words versus 65-85 for other methods) reduce per-email token costs. Additionally, the inverse relationship between diversity and detectability suggests that investing in more sophisticated generation methods, such as BARE, not only improves linguistic variety but also produces data that better mimic human communication patterns.

5.1.7.2. Model-Level Performance Analysis

Beyond method-specific considerations, the choice of the LLM itself has a significant impact on the quality of synthetic data. Our evaluation reveals distinct performance profiles that guide practical deployment decisions.

Claude: Efficiency and Balance Claude emerges as the most efficient generator, achieving the highest diversity scores within more structured generation methods while producing the most concise outputs. Claude, despite generating the shortest emails across all methods (averaging 38-50 words), simultaneously achieves the highest diversity scores within each generation approach. This efficiency, conveying rich information in comparatively short emails, mirrors real-world business communication. Additionally, Claude maintains the most balanced sentiment distribution across all methods, avoiding the extreme positivity bias exhibited by other models. However, this authenticity comes with a trade-off: Claude-generated emails are the most detectable (with up to 70% accuracy), which limits applications that require indistinguishability from human content.

GPT4: Adaptability with Authenticity Trade-offs GPT4 demonstrates adaptability, performing well in scenarios with minimal guidance where other models struggle. Under zero-shot and few-shot prompting, GPT4 achieves the highest distinct-n scores. Most notably, GPT4 produces the hardest-to-detect synthetic emails (46-64% detection accuracy), making it a candidate for applications requiring human-like authenticity. However, this comes at the cost of excessive positivity bias since GPT4 rarely generates negative sentiment. Moreover, the model also favors verbosity, generating longer emails that increase generation costs.

DeepSeek: High Variability Across Methods DeepSeek exhibits the highest variance in performance across generation methods. While performing adequately under structured approaches like BARE and AttrPrompting, it experiences diversity collapse under minimal guidance. This considerable sensitivity suggests DeepSeek requires explicit context to access its linguistic capabilities. Conversely, DeepSeek's wordy zero-shot emails (averaging 86 words) coincide with the lowest diversity scores (distinct-1: 0.005).

Gemini and Mistral: Stable Mid-Tier Performance Gemini and Mistral occupy a middle tier, offering predictable but unremarkable performance. Gemini exhibits similar patterns to DeepSeek, with substantially reduced diversity under minimal prompting, albeit to a lesser extent. It maintains moderate performance across all metrics without excelling in any particular dimension. On the other hand, Mistral demonstrates the most consistent performance across methods but rarely achieves top scores in any metric. While both models demonstrate moderate performance, Gemini's free API tier during our evaluation period provides a compelling rationale for its selection in cost-sensitive scenarios.

5.2. Text Mining

Having established the quality and characteristics of the synthetic data, this section now evaluates its utility for the downstream task of maritime entity extraction. The evaluation begins with an in-depth examination of span-based GLiNER models, exploring a range of pretraining and finetuning strategies. We then assess a generative, template-filling approach using the Qwen-0.5B model. The section concludes with a comparative analysis of these two architectures to identify their respective strengths and limitations for the maritime domain.

5.2.1. GLiNER: Span-based Entity Recognition

5.2.1.1. Encoder Pretraining Analysis

To understand how different pretraining strategies affect downstream GLiNER performance, we start by evaluating the quality of learned entity representations through two complementary approaches. First, we analyze clustering metrics and visualizations to assess the separation of entity types. Second, we evaluate performance on a subset of MTEB (Muennighoff et al., 2022) to measure broader semantic understanding.

Clustering Metrics Table 5.6 presents a quantitative evaluation of the embedding quality across five encoder variants on the OOD NER datasets described in Section 3.4.3. The baseline DeBERTa-v3-small model exhibits poor clustering performance with a negative average Silhouette Score, indicating that entity embeddings often fall closer to incorrect cluster centroids.

Dataset	Silhouette Score					Calinski-Harabasz Index				
	Base	CT-BGE	CT-SBERT	SimCSE	MLM	Base	CT-BGE	CT-SBERT	SimCSE	MLM
Movie	-0.0397	-0.0266	-0.0398	**-0.0163**	-0.0550	55.26	**83.74**	71.97	36.35	50.82
Politics	0.0077	**0.0959**	0.0889	0.0318	-0.0026	89.11	**334.99**	310.59	97.99	87.55
Music	-0.0583	**0.0482**	0.0275	0.0017	-0.0515	34.71	**152.09**	120.88	43.18	33.68
Restaurant	-0.0373	**0.0591**	0.0541	0.0233	-0.0304	33.23	**97.15**	85.15	40.98	33.89
Science	-0.0566	**0.0001**	-0.0242	-0.0070	-0.0703	30.80	**154.34**	123.54	51.22	32.70
Literature	-0.0330	**0.0447**	0.0261	0.0049	-0.0336	24.29	**131.02**	106.35	30.41	24.72
AI	-0.0700	-0.0150	-0.0347	**-0.0137**	-0.0904	17.09	**71.73**	53.77	23.97	16.37
Average	**-0.0410**	**0.0295**	**0.0140**	**0.0035**	**-0.0477**	**40.64**	**146.44**	**124.46**	**46.30**	**39.96**

Table 5.6.: Per-dataset clustering evaluation results comparing five encoder pretraining strategies. **Base**: DeBERTa-v3-small with mean pooling. **CT-BGE**: Contrastively finetuned using bge-large-en-v1.5 similarity targets on entity level. **CT-SBERT**: Contrastively finetuned using all-MiniLM-L6-v2 similarity targets on entity level. **SimCSE**: SimCSE contrastive finetuning on sentence level. **MLM**: Masked language modeling finetuning on the maritime corpus. Silhouette Score measures intra-cluster cohesion versus inter-cluster separation (range: -1 to 1). Calinski-Harabasz Index quantifies cluster definition through variance ratios. Higher values indicate better clustering quality for both metrics. Best scores per dataset are highlighted in **bold**.

Among the tested approaches, entity-contrastive pretraining with BGE embeddings demonstrates the highest efficacy for clustering. This method transforms the baseline's overlapping clusters into well-separated ones, increasing both the Silhouette Score and the CH Index by a substantial margin. The improvement is particularly large in domains with semantically distinct entity types such as *Politics* (0.0959

Silhouette, 334.99 CH Index) and *Restaurant* (0.0591 Silhouette, 97.15 CH Index). These domains benefit from clear categorical boundaries that align well with contrastive separation objectives. The SBERT-based contrastive variant follows a similar pattern but with slightly weaker improvements. Since both entity-contrastive variants use identical training procedures, this performance gap likely reflects differences in the quality of similarity targets. The superior performance of BGE embeddings may originate from their larger scale (335M parameters) compared to SBERT (22M parameters), consistent with the scaling laws described by Kaplan et al. (2020).

In contrast to entity-level methods, SimCSE's sentence-level contrastive training yields modest improvements in clustering. Nevertheless, it achieves the best performance on the *Movie* and *AI* datasets, domains containing abstract, context-dependent entities, where terms like *plot* or *algorithm* require a broader contextual understanding. This suggests that sentence-level learning captures contextual nuances more effectively than naive entity contrasting.

The MLM finetuning objective proves counterproductive, degrading embedding quality across all domains. This deterioration can be attributed to two key factors. First, training the model exclusively on maritime texts narrows its representational capacity for general entity relationships. Second, the MLM objective optimizes for masked token prediction without any explicit mechanism to learn discriminative entity representations, making it fundamentally misaligned with the clustering task.

Embedding Visualization The effectiveness of different pretraining strategies becomes visually apparent through t-SNE projections. Figure 5.7 demonstrates how entity-level contrastive learning excels in domains with clear categorical boundaries. For the Politics dataset, BGE embeddings produce well-separated clusters with semantically coherent overlaps occurring only between closely related types (e.g., *politician* and *person*). SimCSE, by comparison, shows less distinct separation due to its sentence-level training focus.

However, this pattern reverses when looking at more complex domains. The AI dataset (Figure 5.8) presents an apparent paradox. Despite producing visually overlapping clusters, SimCSE achieves superior clustering metrics (measured by Silhouette Score and CH Index) compared to entity-level methods, revealing the fundamental challenge of AI entity classification. Consider the term *network*; it might denote a neural network architecture, a computer network, or a social network, depending on the context. Similarly, *model* could refer to a machine learning model, a

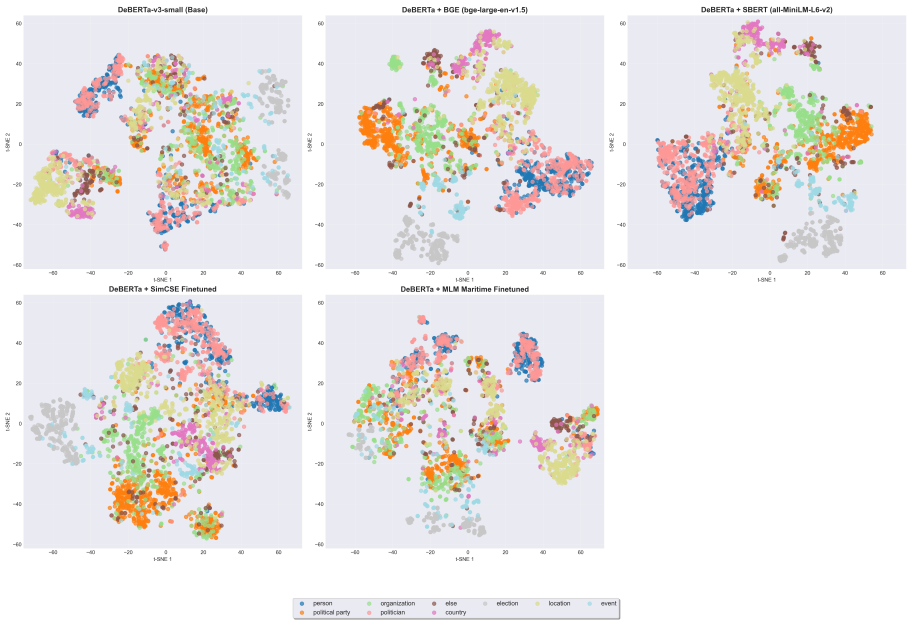

Figure 5.7.: t-SNE visualization of entity embeddings for the *Politics* dataset from MIT (Liu et al., 2021b), comparing baseline DeBERTa-v3-small with contrastive and MLM finetuned variants

database schema model, a 3D graphics model, or a mathematical model of a system. SimCSE's advantage stems from its sentence-level contrastive training on NLI data, which better captures the linguistic context needed to distinguish these polysemous terms.

The limitations of current approaches become most evident in domain-specific tasks. Figure 5.9 shows that neither contrastive nor MLM objectives can successfully disambiguate maritime entities. Vessels and ports remain intermixed across all model variants, with no clear boundaries emerging. t-SNE embedding visualizations for additional datasets are shown in the Appendix C.

Massive Text Embedding Benchmark (MTEB) While clustering metrics evaluate entity-type separation, they may not fully predict downstream task performance. To assess broader utility, we evaluate our pretrained encoders on a subset of MTEB (Muennighoff et al., 2022), which provides a comprehensive benchmark across diverse embedding tasks.

The MTEB results presented in Table 5.7 show a notable reversal from our clustering analysis. SimCSE dominates across all evaluated tasks, with particularly strong performance on retrieval and STS tasks, where it consistently achieves Spearman correlations above 0.6, while entity-contrastive methods barely exceed 0.5. The entity-contrastive approaches show modest improvements over baseline, with BGE

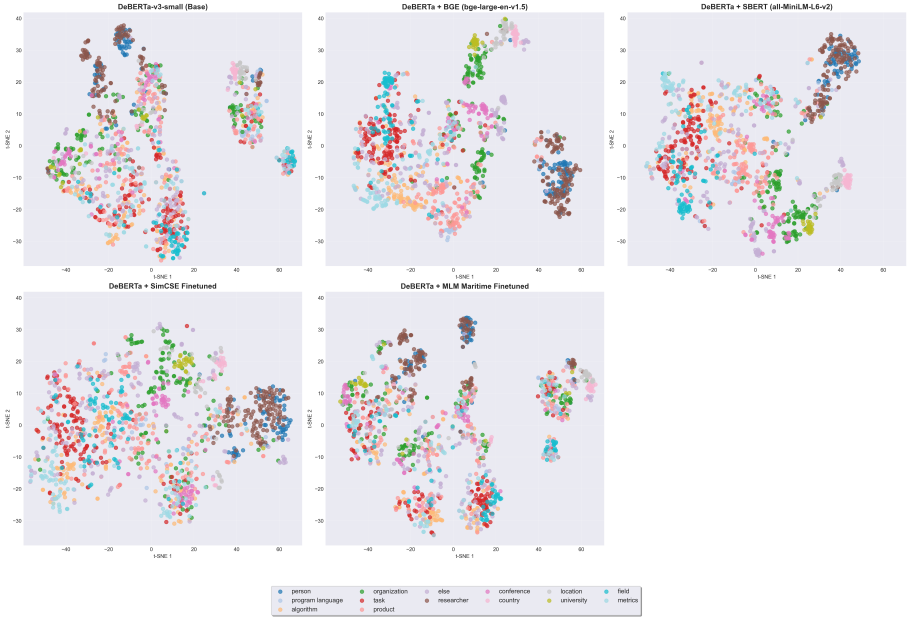

Figure 5.8.: t-SNE visualization of entity embeddings for the *AI* dataset from CrossNER (Liu et al., 2021b). SimCSE (bottom left) shows notably balanced cluster distributions compared to other methods, suggesting its sentence-level training better captures the contextual relationships between abstract AI entities.

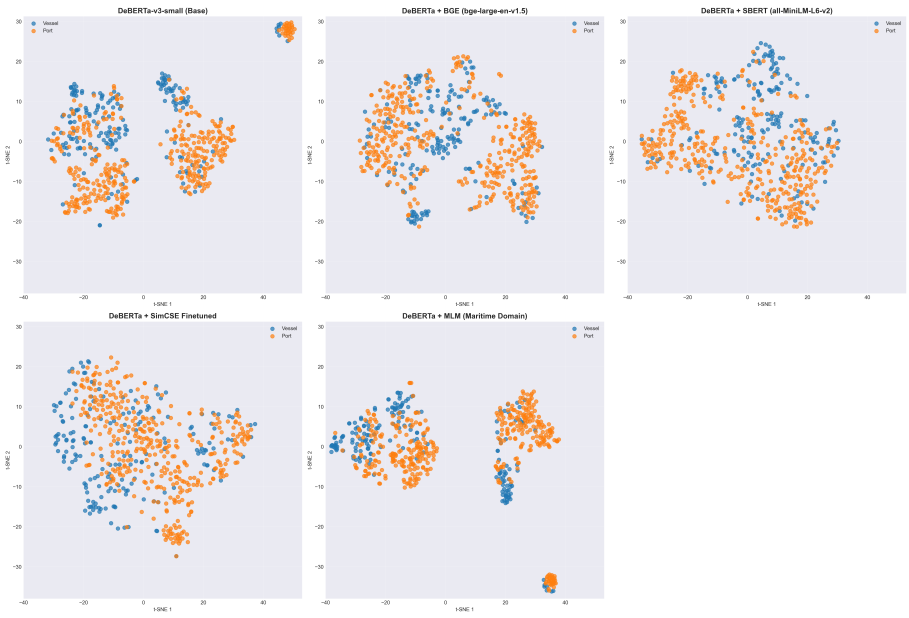

Figure 5.9.: t-SNE visualization of entity embeddings for *ports* and *vessels*, comparing baseline DeBERTa-v3-small with contrastively finetuned variants

Category	Task	Metric	Baseline	CT-SBERT	CT-BGE	SimCSE	MLM
STS	BIOSSES	Spearman	0.3214	0.3092	0.3117	**0.6386**	0.3472
	SICK-R	Spearman	0.4901	0.5387	0.5385	**0.7883**	0.4368
	STS12	Spearman	0.0195	0.2773	0.3776	**0.7192**	0.1838
	STS16	Spearman	0.3472	0.5652	0.5505	**0.7598**	0.4355
	STSBenchmark	Spearman	0.2909	0.4742	0.4791	**0.8061**	0.3297
	STS22	Spearman	0.3133	0.4096	0.4551	**0.6342**	0.3538
Classification	Banking77	Accuracy	0.0130	0.2488	0.2650	**0.5828**	0.0130
	AmazonCounterfactual	Accuracy	0.5403	0.5527	0.5888	**0.6958**	0.5561
Clustering	TwentyNewsgroups	V-Measure	0.0745	0.2198	0.2391	**0.3099**	0.0938
Retrieval	FiQA2018	NDCG@10	0.0009	0.0045	0.0042	**0.0484**	0.0012
Reranking	AskUbuntuDupQuestions	MAP	0.4396	0.4519	0.4478	**0.5027**	0.4205

Table 5.7.: MTEB evaluation results comparing five pretraining strategies on selected tasks from different categories. **Baseline**: DeBERTa-v3-small with mean pooling. **CT-SBERT**: Entity-level contrastive training using all-MiniLM-L6-v2 similarity targets. **CT-BGE**: Entity-level contrastive training using bge-large-en-v1.5 similarity targets. **SimCSE**: Sentence-level contrastive learning following Gao et al. (2021). **MLM**: Masked language modeling finetuned on maritime corpus. All variants use the same DeBERTa-v3-small base architecture. Tasks span semantic textual similarity (STS), classification, clustering, retrieval, and reranking categories. Metrics include Spearman correlation for STS tasks, accuracy for classification, V-measure for clustering, NDCG@10 for retrieval, and MAP for reranking. SimCSE achieves best performance across all evaluated tasks (highlighted in **bold**).

generally outperforming SBERT. Consistent with our clustering analysis, MLM pretraining provides minimal benefit, often performing worse than the baseline model.

These contrasting results highlight a fundamental trade-off in encoder pretraining. While entity-contrastive training optimizes for distinguishing between entity types (beneficial for clustering), sentence-level contrastive objectives like SimCSE develop broader semantic representations that generalize better to diverse downstream tasks.

5.2.1.2. Out of Domain Performance

Having examined how pretraining strategies affect embedding quality and general-purpose performance, we now evaluate their impact on GLiNER's core task: NER. We benchmark five GLiNER variants on the OOD NER datasets used by Zaratiana et al. (2023) in a zero-shot setting, testing generalization without any finetuning. Table 5.8 presents macro-aggregated F1 scores across seven domains. The results demonstrate that no single pretraining strategy dominates universally, with performance varying significantly across different domains.

The contrastively pretrained models achieve marginally higher average performance (SBERT: 55.77%, SimCSE: 55.75%, BGE: 55.74%) compared to the baseline (54.33%),

Model	AI	Literature	Movie	Music	Politics	Restaurant	Science	Average
GLiNER Base	**51.18%**	60.82%	**43.48%**	66.13%	**68.65%**	32.15%	57.89%	54.33%
BGE	47.07%	63.38%	40.31%	71.20%	66.54%	41.12%	**60.55%**	55.74%
SBERT	49.05%	**64.06%**	38.77%	70.58%	68.31%	40.64%	59.00%	**55.77%**
SimCSE	47.18%	63.47%	41.49%	**71.26%**	66.73%	40.76%	59.33%	55.75%
MLM	41.96%	58.12%	37.27%	65.74%	64.03%	**41.25%**	51.65%	51.43%

Table 5.8.: Overall macro F1 Scores by GLiNER variant and domain for zero-shot NER evaluation. The model names correspond to the encoder pretraining strategies: **GLiNER Base** (no special pretraining), **BGE** and **SBERT** (entity-level contrastive learning), **SimCSE** (sentence-level contrastive learning), and **MLM** (domain-adapted). Best scores per domain are highlighted in **bold**.

confirming that semantic separation, whether at entity or sentence level, provides modest but consistent advantages. As expected from its poor clustering and MTEB performance, the MLM variant shows the lowest average macro F1 score (51.43%).

Surprisingly, however, the MLM model achieves the best performance on the *Restaurant* dataset. Table 5.9 reveals that this unexpected strength stems from positive knowledge transfer rather than domain understanding. The maritime corpus, rich with port names, vessel origins, and trade routes, enhanced the model's sensitivity to geographical patterns, which directly benefits *Location*, *Cuisine*, and *Amenity* extraction. Similarly, exposure to numerical data such as vessel dimensions, cargo weights, and distances during training improved *Price* and *Rating* recognition. Thus, the MLM model succeeds by applying learned geographical and numerical patterns rather than understanding the semantics of restaurants.

Entity	GLiNER Base	MLM	SimCSE	BGE	SBERT
Amenity	35.90%	**39.61%**	17.96%	11.73%	9.00%
Cuisine	23.70%	**41.72%**	37.17%	40.69%	40.64%
Dish	51.55%	50.47%	**60.58%**	54.11%	54.90%
Hours	17.03%	18.56%	**23.71%**	20.69%	20.14%
Location	31.54%	43.22%	45.41%	**46.52%**	44.72%
Price	10.91%	37.68%	41.28%	40.00%	**41.57%**
Rating	17.96%	57.56%	58.06%	**61.88%**	60.95%
Restaurant Name	36.65%	36.78%	38.99%	42.78%	**44.12%**
OVERALL	32.15%	**41.25%**	40.76%	41.12%	40.64%

Table 5.9.: Entity-level F1 scores for the *Restaurant* dataset, comparing the zero-shot performance of GLiNER with different pretrained encoders. The maritime-focused MLM model achieves the highest overall F1 score, demonstrating unexpected strength in extracting concrete entities, such as *Amenity*, *Cuisine*, and *Price*. This suggests a transfer of pattern recognition for geographical and numerical data rather than a semantic understanding of the restaurant domain.

The *Music* domain results (Table 5.10) showcase the strengths of contrastive learning. All three contrastive variants (SimCSE: 71.26%, BGE: 71.20%, and SBERT: 70.58%) substantially outperform the baseline (66.13%). This dataset features semantically similar yet categorically distinct entities, making it ideal for contrastive learning objectives. The models excel at disambiguating closely related concepts, such as distinguishing *person* from *musical artist* or *country* from *location*, demonstrating their ability to learn subtle boundaries between similar categories.

Entity	**GLiNER Base**	**MLM**	**SimCSE**	**BGE**	**SBERT**
album	**78.18%**	55.45%	68.06%	70.23%	67.34%
award	46.24%	49.90%	**62.67%**	46.82%	40.51%
band	**83.11%**	73.01%	76.47%	78.99%	77.73%
country	73.68%	87.68%	91.93%	**93.50%**	91.84%
event	**47.80%**	32.49%	39.30%	43.56%	39.27%
location	74.90%	74.62%	**77.38%**	74.21%	76.92%
music genre	**81.23%**	67.47%	75.03%	74.85%	74.85%
musical artist	43.75%	79.96%	79.36%	81.61%	**82.28%**
musical instrument	50.00%	44.68%	48.78%	51.85%	**51.95%**
organisation	**69.87%**	58.67%	61.10%	61.28%	65.08%
person	27.06%	46.49%	45.96%	50.24%	**52.63%**
song	**73.51%**	56.43%	67.24%	67.93%	66.24%
OVERALL	66.13%	65.74%	**71.26%**	71.20%	70.58%

Table 5.10.: Entity-level F1 scores for the *Music* dataset. Contrastively trained models (SimCSE, BGE, SBERT) excel in this domain, demonstrating an advantage in disambiguating semantically similar but distinct entity types, such as *person* and *musical artist* or *country* from *location*.

However, the *AI* dataset (Table 5.11) exposes that this specialization can become problematic in more abstract domains. While contrastive models maintain advantages for polysemous entities, such as *researcher* versus *person* or *country* versus *location*, they struggle with functionally defined concepts, such as *task* and *algorithm*. These entities derive meaning from their contextual role rather than from their inherent properties. The baseline model, unaffected by contrastive constraints, retains greater flexibility for such context-dependent recognition.

These results reveal fundamental trade-offs in encoder pretraining for GLiNER. Contrastive methods excel when entity types have clear semantic boundaries but falter when distinctions are contextual. Domain-specific pretraining, while generally detrimental, can yield unexpected benefits when structural patterns transfer across domains. The optimal strategy thus depends on the target domain's characteristics, whether semantic properties, contextual roles, or structural patterns define entities. Detailed performance breakdowns for the remaining datasets are provided in Appendix D.

Entity	GLiNER Base	MLM	SimCSE	BGE	SBERT
algorithm	**51.07%**	39.93%	42.31%	42.70%	45.37%
conference	**50.93%**	32.30%	37.50%	39.47%	37.41%
country	60.27%	81.55%	**84.54%**	84.00%	84.00%
field	**58.48%**	40.24%	49.08%	48.82%	48.36%
location	50.85%	55.56%	**68.89%**	63.16%	67.35%
metrics	**50.34%**	35.87%	40.00%	39.80%	43.14%
organization	**62.99%**	54.65%	59.46%	60.25%	57.49%
person	41.04%	43.26%	50.41%	48.44%	**54.08%**
product	**48.83%**	32.95%	32.14%	32.88%	34.30%
program language	**77.52%**	59.26%	68.97%	73.97%	69.93%
researcher	3.47%	44.16%	57.72%	50.00%	**65.63%**
task	**49.08%**	31.44%	33.90%	34.98%	34.69%
university	**88.52%**	71.43%	80.00%	85.25%	86.67%
OVERALL	**51.18%**	41.96%	47.18%	47.07%	49.05%

Table 5.11.: Entity-level F1 scores for the *AI* dataset. The baseline GLiNER model outperforms contrastively trained variants on abstract, context-dependent entities, such as *algorithm* and *task*. This highlights a limitation of contrastive methods in domains that lack clear categorical boundaries. However, the contrastive variants still excel at disambiguating polysemous entities like *person* and *researcher*

5.2.1.3. Maritime Domain Performance

Having established the generalization capabilities of different pretraining strategies, we now turn to their performance on the primary task: maritime entity extraction. This section evaluates both zero-shot and finetuned GLiNER variants on our synthetic email datasets, revealing how domain-specific training transforms model performance.

The overall results, shown in Figure 5.10, underscore the importance of finetuning. While pretraining on general corpora, such as PileNER, provides a foundation, specialized finetuning remains essential for good performance in niche domains like maritime shipping. Every finetuned model dramatically outperforms its zero-shot counterpart.

Figure 5.11 quantifies the impact of finetuning. Zero-shot models achieve a mean macro F1 score of 0.658, while general finetuning on 50,000 maritime samples boosts performance to 0.778. The most effective strategies, however, are memory bank finetuning (0.862) and email-specific finetuning (0.828). While the email-tuned models achieve high scores, their performance is likely biased due to finetuning on the same synthetic distribution as the test set. Thus, the memory bank approach emerges as the most robust strategy, achieving superior performance without exposure to the target data distribution.

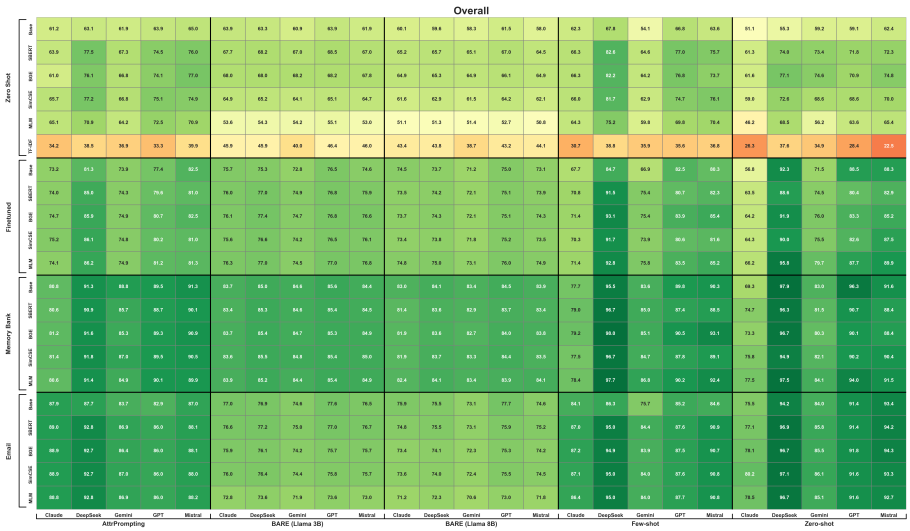

Figure 5.10.: Heatmap of overall macro F1 scores for all GLiNER variants across all data generation configurations. Each cell represents the F1 score of a specific GLiNER model (y-axis) on a specific dataset (x-axis). The y-axis is grouped by finetuning strategy (Zero Shot, Finetuned, Memory Bank, Email), and within each group, by the encoder pretraining method (Base, SBERT, etc.). The x-axis is grouped by the data generation method (e.g., AttrPrompting, BARE). Color coding ranges from red (low performance) to green (high performance). The most prominent trend is the clear performance gap between the zero-shot models (top block) and all finetuned variants, underscoring the benefit of domain adaptation.

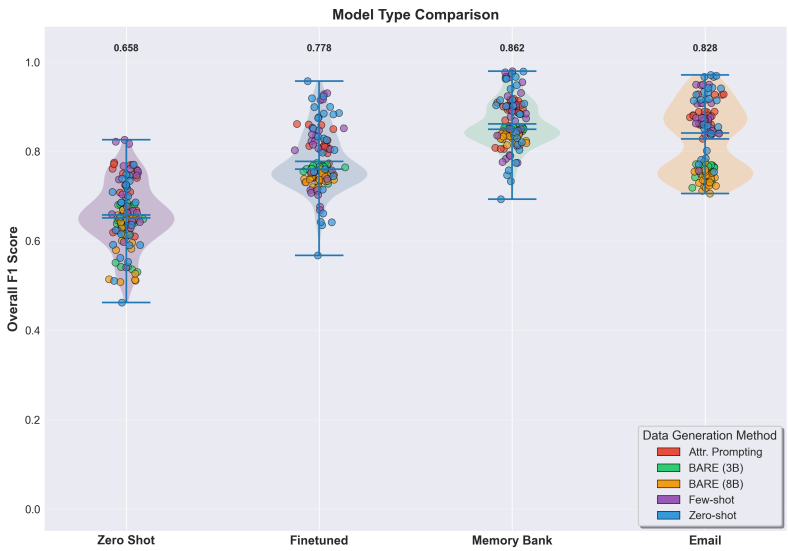

Figure 5.11.: Comparison of overall F1 scores grouped by finetuning strategy. The plot clearly shows the performance hierarchy, with Memory Bank and Email finetuning outperforming general finetuning and the zero-shot baseline. Color coding represents the data generation method.

Contrastive Pretraining Benefits A closer examination of the zero-shot results reveals that contrastive pretraining provides substantial benefits for maritime entity extraction. As shown in the top block of Figure 5.10, the contrastively pretrained variants (SBERT, BGE, SimCSE) consistently outperform the baseline GLiNER across all data generation configurations. The improvements are particularly pronounced on more structured generation methods: for AttrPrompting with Claude, the baseline achieves 61.2% while SimCSE reaches 65.7%. On BARE generations, where the extraction task is most challenging, these gains become even more significant. For instance, on BARE (Llama-8B) with DeepSeek, the baseline GLiNER achieves 67.8%, while the SBERT variant reaches 82.6%, representing an improvement of 14.8 percentage points.

This consistent advantage of contrastive pretraining aligns with our earlier findings from the embedding analysis (Section 5.2.1.1), where these methods showed superior entity-type separation. The benefits persist even after domain-specific finetuning, though the performance gap narrows as all variants converge toward higher scores. This suggests that while contrastive pretraining provides a better initialization for entity recognition, maritime-specific training remains essential for optimal performance.

The Role of Data Source and Generation Method Model performance varies significantly with the characteristics of the data. Figure 5.12 reveals that extraction difficulty is inversely correlated with text diversity. Emails generated by DeepSeek prove easiest to extract from (mean F1: 0.82), while Claude's outputs pose the greatest challenge (mean F1: 0.734). This pattern reflects the findings of our diversity analysis. Claude produces the most varied emails, creating a more complex extraction task, while DeepSeek's outputs collapse into repetitive patterns under minimal guidance. The data generation method also affects difficulty in a similar manner. Generation methodologies such as AttrPrompting and few-shot prompting yield more extractable text than the two-stage BARE process. Consequently, this indicates an inverse relationship between diversity and extractability.

Entity-Specific Challenges: Disambiguation and Rarity Entity-level analysis reveals challenges unique to the maritime domain. A critical task involves disambiguating entities with overlapping forms, for example, distinguishing the vessel *Jakarta Express* from the port of *Jakarta*. Figure 5.13 demonstrates that contrastively pretrained models (SBERT, BGE, SimCSE) perform above baseline at this disambiguation in zero-shot settings. This improvement can be attributed to

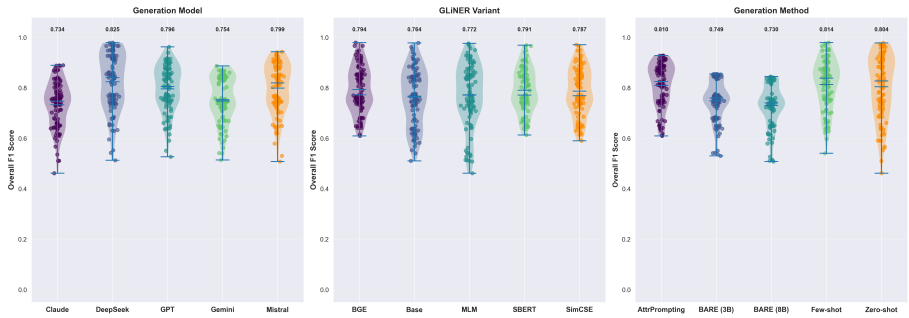

Figure 5.12.: Violin plots showing macro F1 score distributions grouped by (left to right) data generation model, GLiNER encoder variant, and data generation method. The plots show that extraction is most difficult on data from Claude and the BARE method, a finding that aligns with the higher diversity and complexity in their generated text.

their pretraining objective, which optimizes for separating semantically similar concepts.

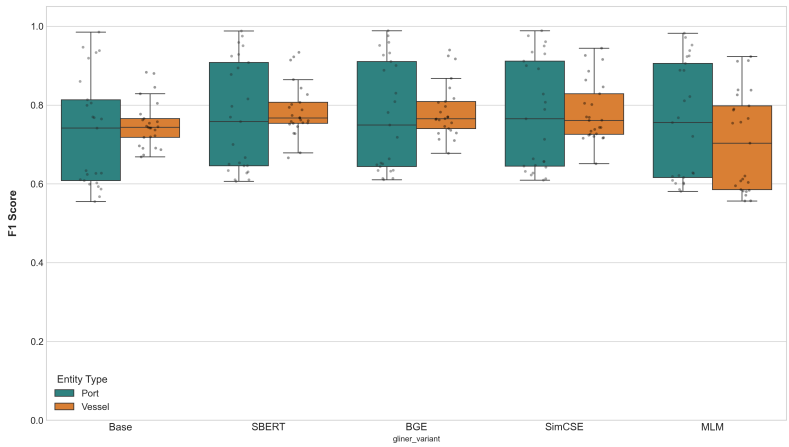

Figure 5.13.: Grouped box plot comparing macro F1 scores for *Port* and *Vessel* entities across zero-shot GLiNER variants. The plot illustrates that contrastively trained models (SBERT, BGE, SimCSE) generally outperform the Base and MLM encoders, confirming their enhanced ability to disambiguate these semantically similar entity types.

The most challenging entity type across all models is Incoterms (Figure 5.14). These standardized trade terms present two distinct difficulties. First, as short acronyms (e.g., CIF, FOB), they can easily be confused with other maritime abbreviations, such as CP (Charter Party) or SHINC (Sundays and Holidays Included). Second, their distribution varies dramatically by generation method. Claude's zero- and few-shot outputs include diverse, realistic maritime terminology that creates ambiguity, whereas minimal guidance in DeepSeek leads to the repetitive use of common terms, such as CIF, with few confounding abbreviations. This explains the heatmap's predominantly low scores, where models struggle most when facing Claude's realistic

mix of maritime terminology, but perform better on simplified distributions, such as those produced by DeepSeek. Ultimately, only email-specific finetuning successfully overcomes this challenge, suggesting that exposure to the full range of maritime abbreviations during training is required for accurate Incoterm extraction.

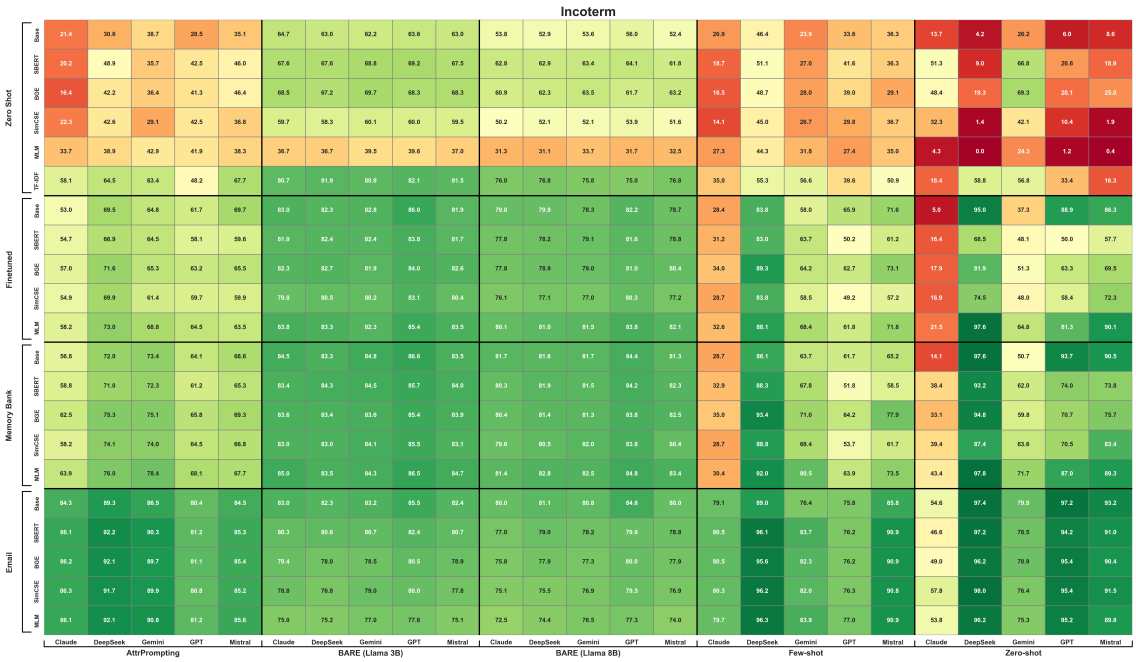

Figure 5.14.: Heatmap of macro F1 scores for the *Incoterm* entity. The universally low scores (predominantly orange and red cells), especially in the zero-shot block, highlight the inherent difficulty of extracting these low-frequency, context-poor acronyms. Performance improves with fine-tuning but remains the lowest of all entity types.

In summary, our in-domain evaluation reveals three key insights. First, finetuning is essential since it enhances performance across all pretraining strategies, with memory bank approaches offering the best balance of effectiveness and robustness. Second, data diversity creates a fundamental trade-off as more realistic synthetic data challenges extraction systems, but is likely necessary to train more capable and generalizable models. Finally, maritime NER presents unique challenges in entity disambiguation and rare-term extraction that require specialized training strategies to overcome.

5.2.2. CodeIE vs. Standard LLM Performance Analysis

The original CodeIE study by Li et al. (2023a) demonstrated that code-specialized models outperform standard LLMs on IE tasks. However, their evaluation was limited to models with more than 175B parameters, a scale that is infeasible for most

local deployments. This raises the question: Do code-specialized models maintain their advantage at smaller scales?

In a limited evaluation covering 24 models, we find that code-based models tend to perform significantly worse than standard LLMs within the 80M to 3B parameter range. Displayed in Figure 5.15, we see that standard LLMs outperform code-based peers by a large margin. While both model types exhibit improved performance in terms of the number of parameters, even comparatively large code models, such as StarCoder 2 with 3 billion parameters (Li et al., 2023b), struggle to produce any valid response.

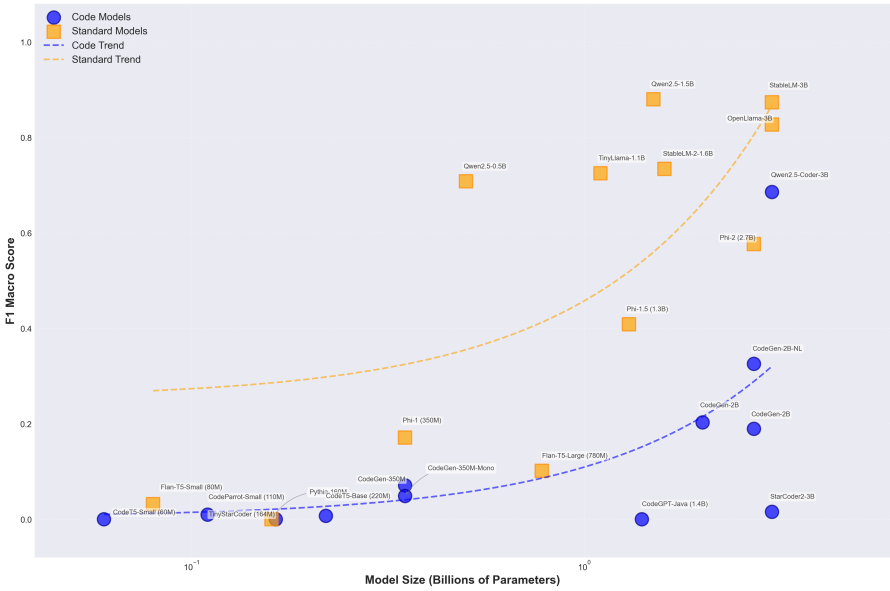

Figure 5.15.: Performance Comparison of Code and Standard LLMs. Standard LLMs (blue) consistently outperform Code LLMs (orange) across all size categories below 3B parameters. The dotted lines represent cumulative mean performance.

This performance difference suggests a capability threshold effect, where code models require a particular number of parameters before their structured reasoning advantages can compensate for deficits in natural language understanding. Code models are primarily trained on programming languages with high structural regularity but limited natural language diversity. For IE tasks, these models must simultaneously comprehend natural language instructions, parse unstructured text, and generate structured outputs —a multifaceted challenge that appears to overwhelm smaller code models. At scales below 3B parameters, code models show two failure patterns. First, they copy the provided examples from the prompt (suggesting pattern recognition without semantic understanding). Second, they return empty extractions (indicating a lack of task understanding). The full results, including precision,

recall, and average inference speed, are shown in the Appendix F.

A noteworthy finding from our analysis is the performance of Qwen-2.5-0.5B (Yang et al., 2024). Despite having only 0.5B parameters, this model achieves an F1 score of 0.708, outperforming both standard and code models with 2-4× more parameters. This competitive extraction performance, achieved at a fraction of the computational cost, leads us to explore Qwen-2.5-0.5B in more detail.

5.2.3. Qwen-2.5-0.5B: Template Filling

The evaluation of Qwen-2.5-0.5B variants reveals distinct performance patterns across different synthetic email generation methods. As illustrated in Figure 5.16, the baseline model (Qwen Base) achieves high F1 scores on AttrPrompting (ranging from 0.770 to 0.907) and few-shot generations (ranging from 0.871 to 0.954), but demonstrates markedly lower performance on BARE-generated content, with F1 scores between 0.396 and 0.450. Across all generation methods, the model variants perform best on DeepSeek-generated emails while encountering the most significant challenges with content generated by Claude.

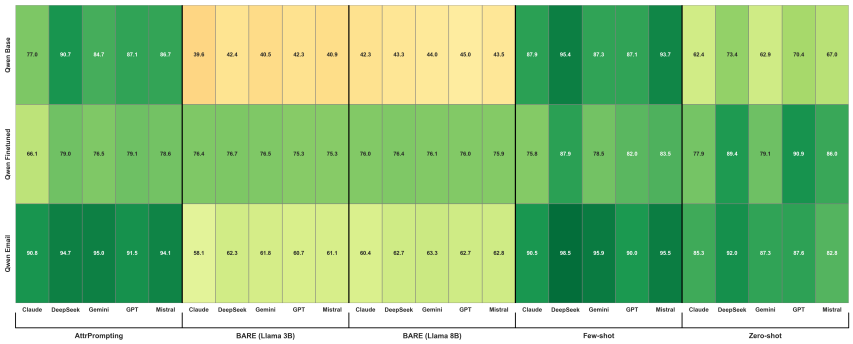

Figure 5.16.: Overall extraction performance heatmap for Qwen model variants across synthetic email generation configurations. Rows represent the three extraction models: Qwen Base (pretrained), Qwen Finetuned (finetuned on 50,000 maritime documents), and Qwen Email (finetuned on 5,000 email chains). Columns show combinations of generation methods and generation models. Cell values indicate macro-averaged F1 scores as percentages, color-coded from red (poor performance) to green (excellent). The heatmap reveals three distinct patterns: (1) Qwen Base performs well on AttrPrompting and few-shot generations but fails on BARE generations (39.6-45.0%), (2) Qwen Email shows universal improvements with the strongest gains on BARE and zero-shot, and (3) Qwen Finetuned trades performance on AttrPrompting and few-shot generations for BARE specialization, achieving the highest BARE scores (75.3-76.7%) among all variants.

The baseline model exhibits varying performance across different entity types, as illustrated in Figure 5.17. While commodity extraction achieves the highest and most consistent performance (mean F1 = 0.847) with minimal variance, and port extraction also performs reliably (mean F1 = 0.777), the model struggles with vessel names and Incoterms. Vessel extractions exhibit a bimodal distribution, with F1 scores clustering at both extremes, indicating that some datasets achieve near-perfect extraction, while others fail entirely. Incoterm extractions prove even more challenging, showing both the lowest mean performance (F1 = 0.386) and the highest variance. The Incoterm extraction scores are distributed across the entire spectrum from near-zero to almost perfect. Consequently, this reveals a lack of domain understanding of maritime terminology, at least when it comes to vessel names and Incoterms.

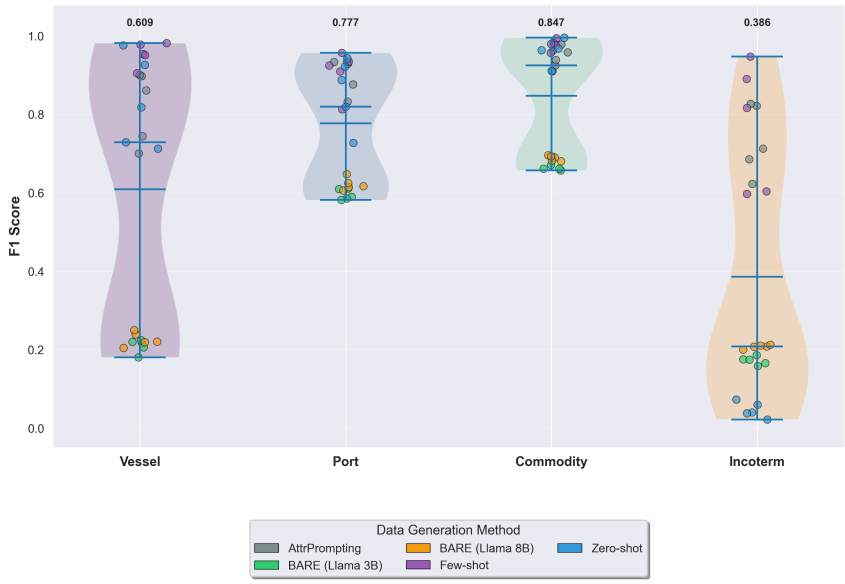

Figure 5.17.: Entity extraction performance distribution for Qwen Base model across four maritime entity types. Each violin plot shows the distribution of macro-averaged F1 scores, with individual dots representing performance on specific dataset configurations (combinations of generation methods and generation models). The horizontal lines indicate the median and mean (with numerical values shown above each subplot). Commodity extraction demonstrates the highest and most consistent performance (mean F1 = 0.847), while incoterm extraction shows the poorest and most variable results (mean F1 = 0.386). Color coding represents the data generation method, showing the difficulty of Qwen Base with BARE generations.

We show the model/method-generation specific extraction performance on Incoterms and vessel names in Figure 5.18 (Incoterms) and Figure 5.19 (Vessels). In line with the overall performance, extractions are generally the hardest from emails generated

through BARE.

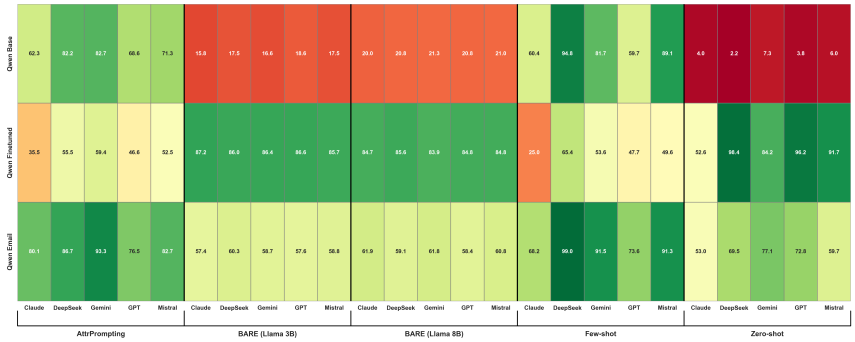

Figure 5.18.: Heatmap of Incoterm extraction performance across different Qwen model variants and synthetic email generation configurations. Rows represent the three extraction models (Qwen Base, Qwen Finetuned, and Qwen Email), while columns show combinations of generation methods and generation models. Cell values show macro-averaged F1 scores as percentages, with color coding from red (poor performance) to green (perfect performance). Qwen Base demonstrates strong performance on attribute prompting, few-shot, and zero-shot, but fails on BARE generations (15.8-21.0%). In contrast, Qwen Finetuned shows strong improvement on BARE performance (84.7-87.2%) and zero-shot generations at the cost of reduced performance on other generation methods. Qwen Email shows improvement on all generation methods, with lower improvement on BARE compared to Qwen Finetuned.

Our two finetuning strategies yield divergent outcomes as shown in Figure 5.16. On the one hand, email-based finetuning improves performance across all generation methods, with the largest increase observed in BARE and zero-shot generations by roughly 20 percentage points each. In comparison, finetuning on the broader maritime dataset (comprising Wikipedia articles, news articles, and template generations) sacrifices performance on attribute prompting and few-shot generations while leading to the best performance on BARE generations, achieving approximately a 35 percentage point improvement over the baseline. Given these findings, both the difficulty of BARE generations and the surprising effectiveness of broader corpus finetuning, we examine BARE extractions in greater detail in the following section.

5.2.3.1. The BARE Challenge: Entity Co-Occurrence

We found that the challenge posed by BARE-generated emails originates from their high variability in entity co-occurrence patterns. Unlike attribute-prompted generations, where all entities typically appear together in a generated email, BARE

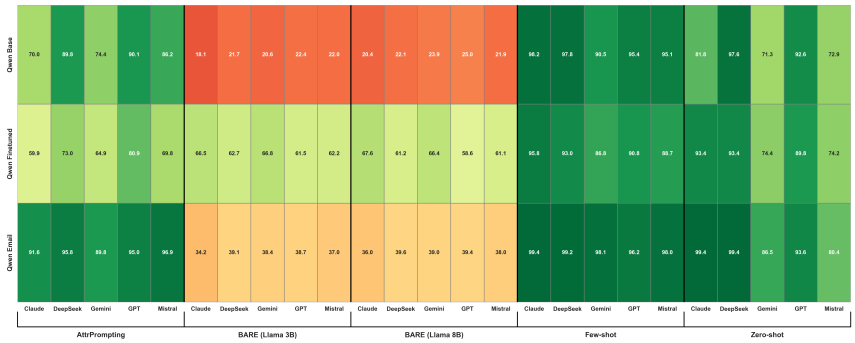

Figure 5.19.: Heatmap of vessel extraction performance across different Qwen model variants and synthetic email generation configurations. Rows represent the three extraction models (Qwen Base, Qwen Finetuned, and Qwen Email), while columns show combinations of generation methods and generation models. Cell values show macro-averaged F1 scores as percentages, with color coding from red (poor performance) to green (perfect performance). Qwen Base achieves strong performance on AttrPrompting (70.0-90.1%), few-shot (87.8-98.2%), and zero-shot (71.3-97.6%) but performs poorly on BARE generations (18.1-25.0%). Qwen Finetuned demonstrates more balanced performance with significantly improved BARE extraction (58.6-67.6%) and lower performance on the remaining generation methods. Qwen Email shows improvement on all generation methods, with lower improvement on BARE compared to Qwen Finetuned.

emails show variations. Some emails generated through BARE contain vessel names and ports without any Incoterms, while others specify only an Incoterm and commodity without mentioning any vessel. This explains why the variant finetuned on the broader maritime corpus performs better on BARE generations but sacrifices performance on the remaining generations. The co-occurrence variance aligns better with the entity distribution found in the broader maritime corpus, where specialized texts, such as Wikipedia articles about specific vessels, rarely mention Incoterms and commodities. In contrast, the email training data exhibits more uniform entity co-occurrence, resulting in a mismatch with BARE's diverse patterns.

We can confirm this quantitatively by looking at the precision and recall metrics presented in Table 5.12. These metrics, micro-averaged to better capture performance on imbalanced datasets, reveal fundamental differences between the two finetuning approaches.

The email-finetuned model features low precision (0.143) paired with high recall (0.657) for vessel name extractions on BARE generations, indicating aggressive over-extraction. We found that it frequently generates placeholder values (*Vessel name*, *Ship Name*) or defaults to common training examples (*MV Ocean Star*) when no

vessel exists in the email. In contrast, the broader maritime finetuning achieves a 28.4% improvement in precision (0.184) at the cost of a 33.2% reduction in recall (0.439), indicating that the model has learned to abstain more effectively; i.e., the broader corpus has taught the model that absence can be genuine.

The pattern intensifies for Incoterms extraction, where the email-finetuned model's precision of 0.232 improves to 0.347 under broader maritime finetuning, resulting in a 49.6% increase. The corresponding recall decreased from 0.569 to 0.295 (48.2% reduction). Thus, the email-finetuned model overfits to common Incoterms such as *CFR* and *FOB*, thereby missing less frequent values.

	Vessel Extraction				**Incoterm Extraction**			
	Qwen Email		**Qwen Finetuned**		**Qwen Email**		**Qwen Finetuned**	
Model	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
BARE Llama 3B								
Claude	0.123	0.709	0.188	0.543	0.211	0.602	0.361	0.313
DeepSeek	0.145	0.680	0.189	0.510	0.224	0.533	0.331	0.278
Gemini	0.140	0.715	0.188	0.453	0.204	0.526	0.329	0.275
GPT	0.140	0.700	0.174	0.521	0.217	0.586	0.365	0.310
Mistral	0.124	0.741	0.143	0.457	0.221	0.594	0.286	0.247
BARE Llama 8B								
Claude	0.145	0.652	0.217	0.460	0.255	0.608	0.357	0.328
DeepSeek	0.174	0.587	0.205	0.370	0.233	0.536	0.374	0.311
Gemini	0.159	0.603	0.201	0.321	0.274	0.560	0.358	0.285
GPT	0.163	0.586	0.180	0.403	0.234	0.554	0.381	0.332
Mistral	0.139	0.594	0.152	0.350	0.249	0.588	0.333	0.273
Average	**0.143**	**0.657**	**0.184**	**0.439**	**0.232**	**0.569**	**0.347**	**0.295**
Change (%)	—		+28.4	-33.2	—		+49.6	-48.2

Table 5.12.: Precision-Recall analysis for vessel and Incoterm extraction on BARE generations (micro averaged).

In summary, the precision-recall analysis between the two finetuning variants discloses a trade-off in how the models handle entity absence. The broader maritime dataset exposes the model to texts where entities naturally occur in isolation or are absent, teaching it to distinguish between genuine entity absence and opportunities for extraction. This leads to an improved performance on less predictable generation methods like BARE, while reducing performance on samples where entities co-occur frequently. The exact opposite holds for email-finetuning.

5.2.4. Comparative Analysis of GLiNER and Qwen

Having examined GLiNER and Qwen individually, we now analyze their comparative strengths and limitations across different finetuning regimes and data generation methods.

Figure 5.20 reveals distinct performance patterns. While both models achieve similar mean macro F1 scores in zero-shot settings (GLiNER: 0.658, Qwen: 0.655), their performance distributions differ markedly. GLiNER maintains consistent performance across all data generation methods, whereas Qwen exhibits high variability, excelling on AttrPrompting and few-shot generations but struggling with the diverse BARE content.

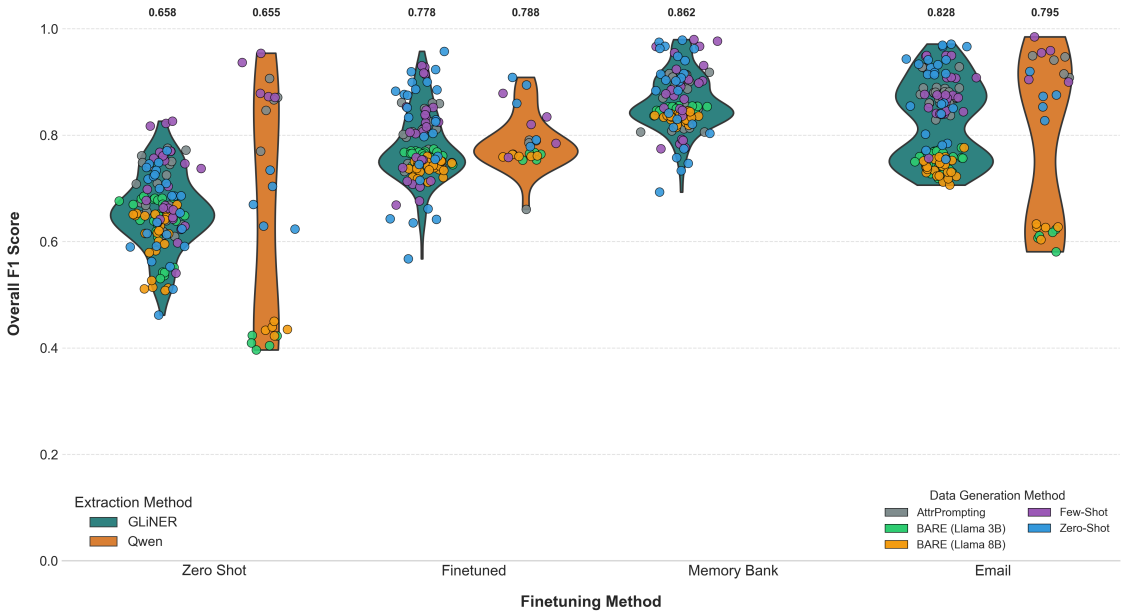

Figure 5.20.: Overall F1 score comparison between GLiNER and Qwen across different experimental conditions. On the x-axis, we present different finetuning methods: **Zero Shot** (no finetuning), **Finetuned** (general maritime corpus), **Memory Bank** (GLiNER-specific contrastive finetuning), and **Email** (finetuning on the target email distribution). Each dot represents the model's performance on a dataset from a specific data generation configuration, with color coding revealing the data generation method. GLiNER consistently shows higher and more stable performance, with the Memory Bank approach achieving the highest overall score (0.862). Qwen's performance is more variable and highly dependent on the alignment of the finetuning data.

Finetuning on the broader maritime corpus produces divergent effects. GLiNER exhibits more uniform improvement across all generation methods, maintaining its narrow performance distribution while increasing the mean performance to 0.778. Conversely, Qwen's finetuning creates a significant redistribution of capabilities. BARE performance improves by approximately 35 percentage points, but at the cost of reduced performance on other generation methods. This trade-off, as discussed in Section 5.2.3.1, can be attributed to the entity co-occurrence patterns in the training data. Overall, finetuning on the broader corpus increases the mean performance of

Qwen to 0.788 and drastically reduces the variance in performance.

Email-specific finetuning elevates both models' performance but reveals bipolar distributions, where BARE generations underperform relative to the other methods, although this effect is more pronounced for Qwen. GLiNER achieves a mean F1 of 0.828 with relatively consistent performance, while Qwen reaches 0.785 with greater variance.

The memory bank approach, unique to GLiNER, achieves the highest overall performance (0.862) while maintaining a very narrow distribution. This consistency across diverse generation methods suggests that contrastive learning with hard negative examples effectively bridges the gap between structured and unstructured text patterns.

Architectural Trade-offs Beyond extraction accuracy, the models differ fundamentally in computational efficiency. GLiNER processes approximately 1,000 emails per minute, while Qwen requires roughly one hour for the same volume. This efficiency gap partially reflects the difference in their parameter counts. GLiNER operates with under 200 million parameters, while Qwen contains 500 million. However, the finetuning strategies also differ significantly. GLiNER's finetuning updates all parameters, potentially limiting generalization to other domains. In contrast, Qwen's PEFT modifies only around 2 million parameters, preserving its general language capabilities while adapting to maritime terminology.

Implications for Maritime IE Overall, these results highlight a fundamental trade-off in maritime entity extraction. GLiNER offers superior consistency, processing speed, and peak performance through specialized finetuning, such as memory banks. Its span-based approach naturally aligns with the bounded nature of maritime entities (vessels, ports, Incoterms), enabling reliable extraction across diverse text patterns. In contrast, Qwen demonstrates the flexibility of generative approaches but suffers from inherent challenges: slower inference, higher variability, and sensitivity to training data distributions. Its strength lies in adaptability, as the same model can handle various tasks beyond entity extraction; however, this generality comes at the cost of reduced performance.

6. Limitations and Future Work

6.1. Limitations

The findings presented in this thesis should be interpreted in light of several methodological limitations. These constraints pertain to the synthetic data generation process, the experimental design, and the data annotation and evaluation procedures.

Constraints in Data Generation and Corpora Our study was subject to two primary data-related constraints. First, hardware limitations restricted our ability to host larger models for the base generations in BARE locally. While the Llama-8B model produced impressive results, findings by Zhu et al. (2025) suggest that larger models could generate even more realistic maritime emails, which might in turn affect the difficulty of the downstream extraction task.

Second, the use of the Enron corpus as a human-authored baseline presents two key limitations. On the one hand, at over two decades old, this dataset may no longer represent contemporary business communication. On the other hand, as one of the few publicly available email datasets, the Enron corpus was likely included in the pretraining data of the models used for the DetectGPT analysis (namely GPT-2 XL and T5-XL). This prior exposure could artificially inflate the likelihood scores assigned to the human-authored text, offering a compelling explanation for the inverted scoring pattern observed in our results. Nevertheless, at the time of this research, the Enron corpus remains the only publicly accessible resource offering the scale and authenticity required for such a comparison, making it a necessary, albeit imperfect, benchmark.

Limitations in Experimental Design Resource constraints also imposed limitations on our experimental design, particularly regarding hyperparameter optimization and the direct comparability of finetuning strategies. An exhaustive hyperparameter search was computationally infeasible. Consequently, our experiments relied on a single configuration for several critical parameters, which may have masked the

optimal performance of certain models and methods. The relative performance differences between pretraining strategies might shift under different hyperparameter configurations, and the observed superiority of certain approaches could be artifacts of suboptimal settings for alternatives.

Data Annotation and Evaluation Concerns Finally, the scale of our synthetic email dataset (approximately 20,000 emails) required an automated labeling process using LLMs rather than manual human annotation. Although we implemented quality control measures, this approach may have introduced inconsistencies, which in turn affect downstream evaluation.

6.2. Future Work

In this thesis, we demonstrate that LLMs can effectively generate maritime email corpora and that specialized extraction models can successfully identify entities from such synthetic data. Building on these dual foundations, future research can advance both the quality of generated data and the accuracy of extraction systems.

Enhanced Generation Through Scale and Innovation Future work should investigate larger base models (70B+ parameters) for BARE implementations, which could capture the nuanced patterns and implicit knowledge characteristic of expert maritime correspondence. Recent advances suggest even greater potential. The hindsight merging technique proposed by Veselovsky et al. (2025) extends BARE by combining base and instruction-tuned models to enhance diversity while maintaining instruction-following capabilities. Applying this approach to maritime email generation could yield superior synthetic corpora that strike a balance between authenticity and diversity. Additionally, domain-specific pretraining of base models on maritime texts or emails before applying generation methods could improve output quality. Finally, systematic prompt engineering also represents an underexplored opportunity, where future research should investigate how prompt design influences the diversity-extractability trade-off, potentially through chain-of-thought prompting or explicit entity co-occurrence modeling.

For extraction models, scaling from DeBERTa-small to medium or large variants, as well as from Qwen-2.5-0.5B to larger parameterized versions or newer model series (Qwen-3), would illuminate how model capacity affects the observed trade-offs between entity clustering and downstream performance. This scaling investigation

should also extend to code-specialized models to test the capability threshold hypothesis identified in our results. While our evaluation found that code models with fewer than 3B parameters were ineffective, future work should explore whether their structured reasoning advantages emerge before the 175B+ parameter scale examined by Li et al. (2023a).

Validation and Benchmarking While our evaluation focused on locally-hosted extraction models for reproducibility and privacy, benchmarking against state-of-the-art remote-hosted LLMs could establish performance ceilings for maritime entity extraction. However, such an assessment must carefully consider privacy implications when processing potentially sensitive business communications. Implementing differential privacy or federated evaluation strategies could enable these comparisons while maintaining data confidentiality.

Domain Generalization Extending our methodology beyond maritime shipping would test the generalizability of observed patterns. Biomedical texts with compositional entities, or legal documents with hierarchical structures, may exhibit different diversity-performance relationships. Such cross-domain validation would determine whether our findings represent fundamental properties of synthetic data generation or domain-specific phenomena.

7. Conclusion

This thesis aimed to address a fundamental challenge in the application of NLP to specialized domains: the scarcity of training data due to concerns over privacy and commercial sensitivity. Through the lens of maritime chartering, a domain where email negotiations drive global trade, we developed and evaluated an end-to-end pipeline from synthetic data generation to IE.

7.1. Revisiting the Research Questions

At the outset of this thesis, we defined two interconnected study objectives that guided our investigation into the generation of synthetic data and its applications in text mining.

Research Question 1: How can we generate high-quality synthetic maritime emails that capture the linguistic characteristics of authentic business correspondence?

Research Question 2: How does the quality and diversity of synthetic training data impact the performance of information extraction systems?

Through systematic experimentation across multiple data generation methods, LLMs, and extraction models, we can now provide empirically grounded answers to these questions.

7.2. Synthetic Data Generation: Methods, Models and Trade-offs

BARE: The Most Effective, Yet Most Sensitive Method Our examination of the first research question revealed that generating authentic maritime emails requires navigating trade-offs between diversity, sentiment, and computational efficiency.

Overall, the BARE methodology emerged as the most effective approach for achieving human-like diversity, with BARE (Llama-8B) under Claude reaching 85% of vocabulary diversity and matching trigram diversity when compared to the human-authored Enron corpus. Ultimately, this two-stage generation process, which leverages base models for creative generation followed by instruction-tuned models for refinement, successfully produces varied linguistic patterns. However, its success is sensitive to implementation details. Our ablation study comparing standard and iterative BARE refinement revealed a critical design choice: sequential refinement degrades quality. Iterative BARE reduced diversity by 30-40%, quadrupled email length, and eliminated negative sentiment.

While BARE achieved the highest diversity metrics among our methods, closer examination of the generated content revealed critical shortcomings of the synthetic email data. First, all generation methods exhibited a positivity bias, with positive sentiment accounting for 50-80% of generated outputs, a finding that aligns with recent studies by Li et al. (2025). At the same time, negative sentiment accounts for less than 10% across all configurations. The refinement step in BARE exacerbates the absence of negative tone, as the refinement model not only neutralizes negative sentiment but also skews the final distribution towards positive sentiment. In contrast, the genuine Enron corpus exhibits substantially more balanced sentiment, with 44.7% positive and 28.3% negative. This systematic under-representation of negative sentiment in our generations reveals a fundamental limitation in current LLMs' ability to produce authentic business communication that includes disagreement, frustration, or conflict. Second, our findings indicate that minimal guidance generation methods are unfeasible when deployed at scale. Zero-shot and few-shot approaches achieved only 5-17% of human-level diversity. This collapse in diversity also affects the entity instances themselves, with zero-shot generations defaulting to the same vessel name (*MV Ocean Star*), incoterm (*CFR*), and commodity (*crude oil*) repeatedly. While such approaches may suffice for rapid prototyping, their limited diversity can lead to model collapse when generating large corpora, underscoring that sophisticated prompting strategies, such as BARE or AttrPrompting, are not merely beneficial but essential.

AttrPrompting: The Trade-off Between Control and Effort As a viable alternative to BARE, AttrPrompting achieves 60% of human-level diversity and provides crucial advantages in control and consistency. Its template-based generation ensures automatic labeling and enforces adherence to the target domain. The primary drawback is the effort required to design these templates, a process that often necessitates domain knowledge and the use of auxiliary data sources. This contrasts with

BARE's more straightforward initiation via a few-shot prompt for the base model, although at the cost of outsourcing the labeling to the LLM as well.

Distinct Model Performance Profiles Model selection proved equally influential on data quality. In our examination, Claude demonstrated the most efficient generation profile, producing the highest diversity scores while maintaining the shortest average email length (38-50 words). This aligns with how businesses strive to communicate, saying more with less. Conversely, GPT4's outputs proved the hardest to detect as AI-generated, but they suffered from excessive positivity bias and verbosity. Finally, DeepSeek showed the highest variance across methods, with its performance collapsing into repetitive patterns under minimal guidance.

An Unexpected Inversion in Authorship Detection Lastly, our authorship detection results revealed an unexpected inversion of the original DetectGPT hypothesis when applied to email data. Different from the findings by Mitchell et al. (2023), we found that human-authored emails were consistently rated as more probable than their perturbations. This is likely due to the presence of the Enron corpus in the training data of both the scoring and rephrasing models, as well as the specificity of the maritime vocabulary used in our generation pipeline. Nevertheless, the adapted detection algorithm could still distinguish synthetic emails with accuracies ranging from 46% to 70%, with more sophisticated generation methods proving more challenging to detect.

In summary, our results demonstrate that while the BARE methodology can generate synthetic emails with near-human levels of diversity, no single approach is without drawbacks. We found a consistent positivity bias across all models, a distinct performance profile for each LLM, and unanticipated challenges in authorship detection. Overall, these findings underscore a central conclusion for data generation: the path to high-quality synthetic data involves an optimization problem that balances diversity, authenticity, control, and cost.

7.3. The Diversity-Extractability Relationship

Our second research objective revealed a fundamental insight: the highest-quality synthetic data, as measured by linguistic diversity, creates the most challenging IE task. This was consistently observed across all our experiments. For instance, in GLiNER, extractions were most difficult from the diverse emails generated by Claude

(mean F1: 0.734), while the more repetitive outputs from DeepSeek proved easiest (mean F1: 0.82). The same pattern was observed for the generation methods, where BARE datasets proved substantially more challenging to parse for both GLiNER and Qwen models.

We found that the mechanism behind the diversity-extractability relationship extends to entity co-occurrence patterns. While entities in AttrPrompting generations typically appear together, BARE produces more realistic variations where some emails feature just subsets of the entity types. This variability proved particularly challenging for Qwen, whose baseline model achieved F1 scores of 0.77-0.91 on AttrPrompting but collapsed to 0.40-0.45 on BARE.

Pretraining Strategies: The Limits of Domain Adaptation Our evaluation of different encoder pretraining strategies showed both promise and limitations. Entity-contrastive pretraining significantly improved clustering metrics (increasing Silhouette Score from -0.041 to 0.030 for BGE embeddings), while SimCSE dominated on MTEB benchmarks. In zero-shot maritime extraction settings for GLiNER, contrastive variants demonstrated substantial improvements, reaching up to 14.8 percentage points on challenging BARE generations, and proved effective at disambiguating overlapping entities. Yet these advantages diminished with finetuning, as all variants converged toward similar performance levels. Surprisingly, MLM pretraining on maritime texts proved counterproductive, degrading both embedding quality and extraction performance despite its domain adaptation. This demonstrates that pretraining alone cannot solve the fundamental challenges of maritime entity extraction, though contrastive pretraining does provide a superior initialization. Moreover, the contrastive GLiNER variants demonstrated their generalization capabilities beyond maritime applications, achieving modest but consistent performance gains on out-of-domain NER datasets (SBERT: 55.77%, SimCSE: 55.75%, BGE: 55.74% versus 54.33% baseline).

Architectural Trade-offs and Finetuning Strategies Furthermore, we showed fundamental differences between span-based and generative approaches. GLiNER's span-based architecture proved more suitable for maritime entity extraction, achieving the highest overall performance (0.862 F1) through contrastive finetuning with hard negative mining and data augmentation while processing 1,000 emails per minute. In contrast, Qwen's generative approach, despite its flexibility and parameter efficiency (only 2M parameters updated via PEFT), suffered from 60× slower inference and higher sensitivity to training data distributions. While both models

exhibited similar baseline performance in zero-shot settings (F1 scores of 0.658 and 0.655, respectively), their responses to finetuning diverged dramatically. GLiNER demonstrated consistent improvement across all generation methods. Qwen, however, showed a redistribution of capabilities: finetuning on the broader maritime corpus improved BARE performance by 35 percentage points but degraded performance on other data generation methods. We found this trade-off directly related to entity co-occurrence patterns. The broader maritime corpus featured more realistic entity distributions where vessels, ports, and commodities appeared independently, aligning better with BARE's variable patterns than with the uniform co-occurrence found in AttrPrompting or few-shot generations.

The Capability Threshold Effect in Code Models Finally, contrary to Li et al. (2023a), who demonstrated superior performance in IE of code-specialized models at scales exceeding 175B parameters, we found that code models catastrophically fail below 3B parameters. Even relatively large code models struggled to produce valid extractions. We hypothesize a capability threshold effect: code models require sufficient scale to overcome their limited exposure to natural language before their structured reasoning advantages manifest. Below this threshold, their training on syntactically rigid programming languages becomes a liability rather than an asset.

7.4. Closing Remarks

As specialized domains increasingly seek to leverage NLP while preserving confidentiality, this thesis demonstrates that synthetic data generation offers a viable path forward. Our work uncovered three fundamental relationships that reshape our approach to synthetic data generation. First, the diversity-extractability relationship reveals that higher quality synthetic data, as measured by diversity, creates more challenging extraction tasks. Second, we confirmed a systematic positivity bias across all LLMs, with negative sentiment consistently under-represented and refinement processes exacerbating this limitation. This poses particular challenges for generating realistic business communications that include conflict or disagreement. Third, we demonstrated how entity co-occurrence patterns serve as a critical dimension when developing domain-specific extraction models.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv:2303.08774*.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv:1707.02919*.

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 1:1.

Samuel A Assefa, Danial Dervovic, Mahmoud Mahfouz, Robert E Tillman, Prashant Reddy, and Manuela Veloso. 2020. Generating synthetic data in finance: Opportunities, challenges and pitfalls. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–8.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: Analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Tadeusz Caliński and Jerzy Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.

Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I Abdin. 2024. On the diversity of synthetic data and its impact on training large language models. *arXiv:2410.15226*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR.

Parag Pravin Dakle and Dan Moldovan. 2020. CEREC: A corpus for entity resolution in email conversations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 339–349, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *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)*, pages 4171–4186.

Kelvin Du, Yazhi Zhao, Rui Mao, Frank Xing, and Erik Cambria. 2025. Natural language processing in finance: A survey. *Information Fusion*, 115:102755.

Irmina Durlik, Tymoteusz Miller, Ewelina Kostecka, Adrianna Łobodzińska, and Tomasz Kostecki. 2024. Harnessing AI for sustainable shipping and green ports: Challenges and opportunities. *Applied Sciences*, 14(14):5994.

Maël Fabien, Esau Villatoro-Tello, Petr Motlicek, and Shantipriya Parida. 2020. BertAA: BERT fine-tuning for authorship attribution. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 127–137.

Siyavash Filom, Amir M Amiri, and Saiedeh Razavi. 2022. Applications of machine learning methods in port operations–a systematic literature review. *Transportation Research Part E: Logistics and Transportation Review*, 161:102722.

United Nations Economic Commission for Europe. 2025. UN/LOCODE code list by country and territory. Accessed: 4 June 2025.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv:2101.00027*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, page 6894. Association for Computational Linguistics.

Pierre-Yves Genest, Pierre Edouard Portier, Előd Egyed Zsigmond, and Martino Lovisetto. 2025. OWNER—Towards unsupervised open-world named entity recognition. *IEEE Access*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv:2407.21783*.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. *arXiv:2306.11644*.

Yanzhu Guo, Guokan Shang, Michalis Vazirgiannis, and Chloé Clavel. 2023. The curious decline of linguistic diversity: Training language models on synthetic text. *arXiv:2311.09807*.

Jochen Hartmann, Mark Heitmann, Christian Siebert, and Christina Schamp. 2023. More than a feeling: Accuracy and application of sentiment analysis. *International Journal of Research in Marketing*, 40(1):75–87.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv:2111.09543*.

Fred Heiding, Simon Lermen, Andrew Kao, Bruce Schneier, and Arun Vishwanath. 2024. Evaluating large language models' capability to launch fully automated spear phishing campaigns: Validated on human subjects. *arXiv:2412.00586*.

Geoffrey E Hinton and Sam Roweis. 2002. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python. https://doi.org/10.5281/zenodo.1212303.

Andreas Hotho, Andreas Nürnberger, and Gerhard Paaß. 2005. A brief survey of text mining. *Journal for Language Technology and Computational Linguistics*, 20(1):19–62.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Baixiang Huang, Canyu Chen, and Kai Shu. 2025. Authorship attribution in the era of llms: Problems, methodologies, and challenges. *ACM SIGKDD Explorations Newsletter*, 26(2):21–43.

Ibrahim. 2023. Global Cargo Ships Dataset. Kaggle Dataset. [CC0: Public Domain], accessed 10 March 2025.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv:2401.04088*.

Kelsey Jordahl, Joris Van den Bossche, Martin Fleischmann, Jacob Wasserman, James McBride, Jeffrey Gerard, Jeff Tratner, Matthew Perry, Adrian Garcia Badaracco, Carson Farmer, Geir Arne Hjelle, Alan D. Snow, Micah Cochran, Sean Gillies, Lucas Culbertson, Matt Bartos, Nick Eubank, maxalbert, Aleksey Bilogur, Sergio Rey, Christopher Ren, Dani Arribas-Bel, Leah Wasser, Levi John Wolf, Martin Journois, Joshua Wilson, Adam Greenhall, Chris Holdgraf, Filipe, and François Leblanc. 2020. geopandas/geopandas: V0.8.1.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv:2001.08361*.

Rabimba Karanjai. 2022. Targeted phishing campaigns using large scale language models. *arXiv:2301.00665*.

Bryan Klimt and Yiming Yang. 2004. Introducing the enron corpus. In *CEAS*, volume 4, page 1.

Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, et al. 2024. Longlamp: A benchmark for personalized long-form text generation. *arXiv:2407.11016*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Michal Laclavík, Štefan Dlugolinskỳ, Martin Šeleng, Marcel Kvassay, Emil Gatial, Zoltán Balogh, and Ladislav Hluchỳ. 2011. Email analysis and information extraction for enterprise benefit. *Computing and informatics*, 30(1):57–87.

John Lafferty, Andrew McCallum, Fernando Pereira, et al. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Icml*, volume 1, page 3. Williamstown, MA.

Jean Lee, Nicholas Stevens, and Soyeon Caren Han. 2025. Large language models in finance (FinLLMs). *Neural Computing and Applications*, pages 1–15.

Haoran Li, Qingxiu Dong, Zhengyang Tang, Chaojun Wang, Xingxing Zhang, Haoyang Huang, Shaohan Huang, Xiaolong Huang, Zeqiang Huang, Dongdong Zhang, Yuxian Gu, Xin Cheng, Xun Wang, Si-Qing Chen, Li Dong, Wei Lu, Zhifang Sui, Benyou Wang, Wai Lam, and Furu Wei. 2024a. Synthetic data (almost) from scratch: Generalized instruction tuning for language models.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering*, 34(1):50–70.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv:1510.03055*.

Peng Li, Tianxiang Sun, Qiong Tang, Hang Yan, Yuanbin Wu, Xuanjing Huang, and Xipeng Qiu. 2023a. Codeie: Large code generation models are better few-shot information extractors. *arXiv:2305.05711*.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. 2023b. Starcoder: May the source be with you! *Transactions on Machine Learning Research*.

Weijiang Li, Yinmeng Lai, Sandeep Soni, and Koustuv Saha. 2025. Emails by LLMs: A comparison of language in AI-Generated and human-written emails.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023c. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10443–10461, Singapore. Association for Computational Linguistics.

Zixuan Li, Yutao Zeng, Yuxin Zuo, Weicheng Ren, Wenxuan Liu, Miao Su, Yucan Guo, Yantao Liu, Xiang Li, Zhilei Hu, et al. 2024b. Knowcoder: Coding structured knowledge into llms for universal information extraction. *arXiv:2403.07969*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv:2412.19437*.

Jingjing Liu, Panupong Pasupat, Scott Cyphers, and Jim Glass. 2013. Asgard: A portable architecture for multilingual dialogue systems. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8386–8390. IEEE.

Xiao-Yang Liu, Guoxuan Wang, Hongyang Yang, and Daochen Zha. 2023. Fingpt: Democratizing internet-scale data for financial large language models. *arXiv:2307.10485*.

Yihe Liu, Anushk Mittal, Diyi Yang, and Amy Bruckman. 2022. Will AI console me when i lose my pet? understanding perceptions of AI-mediated email writing. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pages 1–13.

Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2021a. Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*, pages 4513–4519.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2021b. Crossner: Evaluating cross-domain named entity recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13452–13460.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. *arXiv:2406.15126*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv:1711.05101*.

Yingzhou Lu, Minjie Shen, Huazheng Wang, Xiao Wang, Capucine van Rechem, Tianfan Fu, and Wenqi Wei. 2023. Machine learning for synthetic data generation: A review. *arXiv:2302.04062*.

Andrei Mikheev, Marc Moens, and Claire Grover. 1999. Named entity recognition without gazetteers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv:1301.3781*.

Einat Minkov, Richard C Wang, and William Cohen. 2005. Extracting personal names from email: Applying named entity recognition to informal text. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 443–450.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. MTEB: Massive text embedding benchmark. *arXiv:2210.07316*.

Ziaul Haque Munim, Mariia Dushenko, Veronica Jaramillo Jimenez, Mohammad Hassan Shakil, and Marius Imset. 2020. Big data and artificial intelligence in the maritime industry: A bibliometric review and future research directions. *Maritime Policy & Management*, 47(5):577–597.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Sanjeet Singh Naik. 2023. Shipping Ports Around The World. Kaggle Dataset. [CC0: Public Domain], accessed 10 March 2025.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2022. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv:2203.13474*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv:1807.03748*.

Vishakh Padmakumar and He He. 2023. Does writing with language models reduce content diversity? *arXiv:2309.05196*.

G. Papageorgiou, P. Economou, and S. Bersimis. 2024. A method for optimizing text preprocessing and text classification using multiple cycles of learning with an application on shipbrokers emails. *Journal of Applied Statistics*, 51(13):2592–2626.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Vamsi K Potluru, Daniel Borrajo, Andrea Coletta, Niccolò Dalmasso, Yousef El-Laham, Elizabeth Fons, Mohsen Ghassemi, Sriram Gopalakrishnan, Vikesh Gosai, Eleonora Kreačić, et al. 2023. Synthetic data applications in finance. *arXiv:2401.00081*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Sebastián Ramírez. 2025. FastAPI. https://fastapi.tiangolo.com. FastAPI framework, high performance, easy to learn, fast to code, ready for production. Repository: https://github.com/fastapi/fastapi. License: MIT.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Philippe Remy. 2021. Name dataset. https://github.com/philipperemy/name-dataset.

Peter J Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2023. Gollie: Annotation guidelines improve zero-shot information-extraction. *arXiv:2310.03668*.

Chantal Shaib, Joe Barrow, Jiuding Sun, Alexa F Siu, Byron C Wallace, and Ani Nenkova. 2024. Standardizing the measurement of text diversity: A tool and a comparative analysis of scores. *arXiv:2403.00553*.

Sung-Ho Shin, Oh Kyoung Kwon, Xiao Ruan, Prem Chhetri, Paul Tae-Woo Lee, and Shahrooz Shahparvari. 2018. Analyzing sustainability literature in maritime studies with text mining. *Sustainability*, 10(10):3522.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. 2024. AI models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759.

Saurabh Srivastava, Gaurav Singh, Shou Matsumoto, Ali Raz, Paulo Costa, Joshua Poore, and Ziyu Yao. 2023. Mailex: Email event and argument extraction. *arXiv:2305.13469*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: A family of highly capable multimodal models. *arXiv:2312.11805*.

Mildred C Templin. 1957. Certain language skills in children; their development and interrelationships.

United Nations Conference on Trade and Development. 2024. Review of maritime transport 2024. Technical report, UNCTAD.

Veniamin Veselovsky, Benedikt Stroebl, Gianluca Bencomo, Dilip Arumugam, Lisa Schut, Arvind Narayanan, and Thomas L Griffiths. 2025. Hindsight merging: Diverse data generation with language models. In *The 41st Conference on Uncertainty in Artificial Intelligence*.

VesselFinder. 2025. Vessel tracking overview. Screenshot taken on March 10, 2025.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. Gpt-ner: Named entity recognition via large language models. *arXiv:2304.10428*.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pages 9929–9939. PMLR.

Yue Wang, Weishi Wang, Shafiq Joty, and Steven CH Hoi. 2021. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. *arXiv:2109.00859*.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv:2303.17564*.

Guangnian Xiao, Daoqi Yang, Lang Xu, Jinpei Li, and Ziran Jiang. 2024a. The application of artificial intelligence technology in shipping: A bibliometric review. *Journal of Marine Science and Engineering*, 12(4):624.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024b. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6):186357.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv:2407.10671*.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023a. Fingpt: Open-source financial large language models. *arXiv:2306.06031*.

Yi Yang, Yixuan Tang, and Kar Yan Tam. 2023b. Investlm: A large language model for investment using financial domain instruction tuning. *arXiv:2309.13064*.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2024. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2023. GLiNER: Generalist model for named entity recognition using bidirectional transformer. *arXiv:2311.08526*.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *arXiv:2401.02385*.

Shiyue Zhang, Asli Celikyilmaz, Jianfeng Gao, and Mohit Bansal. 2021. Emailsum: Abstractive email thread summarization. In *Proceedings of the 59th*

Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 6895–6909.

Zikang Zhang, Wangjie You, Tianci Wu, Xinrui Wang, Juntao Li, and Min Zhang. 2025. A survey of generative information extraction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4840–4870.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2023. UniversalNER: Targeted distillation from large language models for open named entity recognition.

Yusheng Zhou, Xueqin Wang, and Kum Fai Yuen. 2021. Sustainability disclosure for container shipping: A text-mining approach. *Transport Policy*, 110:465–477.

Alan Zhu, Parth Asawa, Jared Quincy Davis, Lingjiao Chen, Ion Stoica, Joseph E Gonzalez, and Matei Zaharia. 2025. BARE: Combining base and instruction-tuned language models for better synthetic data generation. *arXiv:2502.01697*.

A. Synthetic Data Generation

This chapter provides a detailed reference for the various prompts and attributes used to generate the synthetic maritime email datasets discussed in Chapter 4. It outlines the specific configurations for four distinct generation strategies: Attr-Prompting, BARE, Zero-Shot Prompting, and Few-Shot Prompting. Each section presents the exact templates and parameters employed to interact with the generating LLMs.

A.1. Attribute Prompting (AttrPrompting)

AttrPrompting Prompting is a highly structured method designed to generate diverse and controlled synthetic data, by systematically populating a template with randomized maritime-specific attributes. In the following subsections, we detail the individual attributes before outlining the prompt template itself.

A.1.1. Email Attributes

A.1.1.1. Vessel Names and Details

A dictionary of vessel information is constructed from the *Global Cargo Ships Dataset* (see Section 3.2.2). These details are injected into the email chain, allowing discussions about vessel suitability and capacity. During prompt generation, one record is chosen randomly. Each entry in the resulting dictionary contains:

- **Name**
  The vessel's name acts as a unique identifier within the dictionary.
- **Deadweight Tonnage (DWT)**
  The cargo-carrying capacity in tons, indicative of the ship's size.
- **Length Overall (LOA)**
  The total length of the vessel in meters.

Sample Vessel Entry

```
{
  "name": "MAASTRICHT MAERSK",
  "dwt": 190326,
  "loa": 399
}
```

A.1.1.2. Port Names

The port names are sourced from the *Shipping Ports Around The World* dataset (see Section 3.2.1). It contains worldwide port names, which are title-cased and stored in a list. Load and discharge ports are chosen randomly, ensuring that the loadport is different from the discharge port.

A.1.1.3. Names

To populate broker and charterer names, the system uses the *NameDataset* Remy (2021), from which the top 250 US-based female first names, top 250 US-based male first names, and 400 US-based last names are extracted. The final first name pool combines both male and female lists, allowing for the random selection of broker and charterer names with increased realism.

A.1.1.4. Incoterms

International Commercial Terms (Incoterms) are widely used in maritime shipping, defining how costs, responsibilities, and risks are allocated between parties involved in the transaction. The attributes include a selection of the most commonly used Incoterms:

- **Free on Board (FOB)**
  The seller is responsible for costs and risks until the goods are on board at the port of departure. The buyer then covers shipping and any subsequent expenses or risks.
- **Cost and Freight (CFR)**
  The seller pays to transport the goods to the named destination port, but the buyer assumes all risk once the goods are loaded on the ship.
- **Cost, Insurance, and Freight (CIF)**

The seller covers transport costs and insurance up to the destination port. Risk transfers to the buyer once the goods are loaded onto the vessel.

- **Delivered at Place (DAP)**
  The seller delivers the goods to a specified destination, with the buyer responsible for import clearance. Risk transfers upon arrival, before unloading.

- **Delivered Duty Paid (DDP)**
  The seller handles all costs and risks, including import duties, until goods arrive at the final destination. The buyer simply receives the goods with no additional formalities.

A.1.1.5. Commodities

One commodity is sampled randomly for each chain from a predefined list to drive negotiation specifics.

- **Crude Oil**
  A naturally occurring, unrefined petroleum product used primarily to produce fuels and other industrial products.

- **Wheat**
  A widely cultivated cereal grain and a staple food source for much of the global population.

- **Palm Oil**
  A versatile vegetable oil derived from the fruit of oil palm trees and used in many foods and consumer goods.

- **Coal**
  A fossil fuel formed from decomposed plant matter, heavily used for electricity generation and industrial processes.

- **Iron Ore**
  The primary source of iron used to make steel and other alloys for construction, and machinery.

- **Liquefied Natural Gas (LNG)**
  Natural gas cooled to a liquid state for efficient storage and transport, commonly used for heating and electricity.

- **Soybeans**
  Protein-rich legumes used for animal feed, cooking oil, and numerous processed

food products.

- **Corn**
  A globally important cereal crop consumed as food, used as livestock feed, and processed into industrial products.
- **Cotton**
  A natural fiber grown for textiles, providing materials for clothing, household goods, and industrial uses.
- **Sugar**
  A sweetener primarily derived from sugarcane or sugar beets, widely used in foods and beverages.
- **Rice**
  A staple grain for over half of the world's population, grown in water-intensive paddies.
- **Aluminum**
  A lightweight, corrosion-resistant metal used across industries like transportation, construction, and packaging.

A.1.1.6. Chartering Abbreviations

Maritime shipping relies heavily on abbreviations. To simulate realistic conversations, we manually curated a list of around 100 such abbreviations. For each email generation, a random subset of up to three abbreviations is inserted into the prompt as an optional set of terms the LLM may use during generation.

A.1.1.7. Charterparty Terms

This collection references common charterparty agreements that outline legal frameworks between the charterer and the owner for vessel hire. A single term is sampled to specify the context under which the negotiation is happening.

- **ABA Tankvoy**
  A short-term tanker charter used for spot trading, often with variable rates.
- **GAFTA**
  A standard charterparty for bulk liquid cargo under the rules of the Grain and Feed Trade Association (GAFTA).

- **FOSFA**
  A charterparty under the Federation of Oils, Seeds and Fats Associations, commonly used for the transport of edible oils.
- **Gencon**
  A widely used general-purpose charterparty, particularly for single voyages in the dry bulk shipping sector.
- **Bareboat Charter**
  An arrangement where the vessel is provided without crew, fuel, or insurance, and full operational control is transferred to the charterer.
- **Time Charter**
  An arrangement where the vessel is hired for a specific period, but the owner retains operational management, including providing the crew and maintaining the vessel.

A.1.1.8. Brokers

A small set of fictional brokerage firms is maintained, including *Anderson Shipping Brokers* and *Global Maritime Brokers*, among others. One is selected per chain to denote the broker's organizational affiliation.

A.1.1.9. Formality Level

Since emails can come in various forms, three distinct formality levels are defined:

- **Formal**
  Used for official or high-level communications, typically featuring a polite and carefully structured tone.
- **Informal**
  More casual with relaxed language, colloquial expressions, and minimal formalities.
- **Colleagues**
  Moderate tone suitable for workplace peers, maintaining professionalism while allowing mild familiarity.

These formality levels influence the use of salutation, degree of politeness, and structure of the conversation.

A.1.1.10. Level of English

This attribute covers five categories, determining the overall linguistic sophistication and potential grammatical or lexical nuances:

- **Expert**
  The writers demonstrates advanced or specialized domain knowledge, using technical terminology with minimal errors.
- **Intermediate**
  The writers has a moderate level of proficiency, occasionally seeking clarifications but overall functioning well.
- **Foreign**
  The writer is non-native, potentially including minor grammatical inaccuracies, borrowed phrases, or direct translations.
- **Business-like**
  The writer employs a professional, concise style, focusing on relevant details and formal structures.
- **Colloquial**
  The writer uses everyday language with informal expressions, contractions, or slang.

A.1.1.11. Verbosity Levels

To capture varying message lengths, five different levels of verbosity are defined:

- **Ultra Concise**
  Contains only the bare essentials, delivering key information in the fewest possible words.
- **Very Concise**
  Adds a small amount of context beyond the essentials while remaining brief.
- **Concise**
  Strikes a balance between brevity and clarity, providing relevant details in short paragraphs.
- **Normal**
  Maintains a moderate level of detail akin to everyday business communication, covering necessary background and clarifications.

- **Detailed**
  Provides thorough explanations and instructions with ample supporting information to ensure a comprehensive understanding.

This influences the amount of detail or explanation included in each email.

A.1.1.12. Type of Noise

To simulate real-world messages, seven types of synthetic noise, ranging from punctuation anomalies and typographic errors to off-topic details, were used. At a probability threshold of 13%, one of these noise types may be introduced into the email generation template, resembling authentic communications.

A.1.1.13. Conversation Mode

One of the most critical aspects of the prompt template is the conversation mode, which defines the overarching theme of the email chain. Each mode dictates:

- Who initiates the first email (broker or charterer).
- Typical negotiation progressions such as discount requests, specification clarifications, or post-deal confirmations.
- Potential outcomes such as agreement, no agreement, or dispute resolution.

A.1.1.14. Anchor Date

Each email chain is anchored to a random date between January 1, 2005, and December 31, 2025. This simulates plausible historical or near-future negotiations. The date is formatted according to one of several formats (e.g., `%Y-%m-%d`, `%d.%m.%Y`) to introduce regional and stylistic variations.

A.1.1.15. Cargo Size

Represents the shipment tonnage, ranging from a minimum of 100 MT to a maximum of 12,500 MT, allowing flexibility for both small and larger loads.

A.1.1.16. Broker Demeanor

- **Uncertain**:
  The broker appears hesitant or less sure about rates and details.
- **Confident**:
  The broker expresses clear certainty and well-supported arguments.

A.1.1.17. Email Count

Indicates the number of emails in the conversation, randomly set between 3 and 10.

A.1.2. Prompt

This section contains the exact prompt template used to instruct the language model in generating synthetic email chains related to maritime contexts. Variable placeholders (shown as {variable_name}) are populated probabilistically from the different attributes outlined above.

```
Generate a synthetic email chain in JSON simulating a realistic maritime shipping
communication between a broker and a charterer. The chain should reflect authentic
back-and-forth exchanges and contain timestamps, short disclaimers (optional), and
shipping jargon.

The communication starts on {anchor_date}.

Essential Attributes:
Broker:                     "{broker}"
Commodity:                  "{commodity}"
Load Port:                  "{load_port}"
Discharge Port:             "{discharge_port}"
Cargo Size:                 "{cargo_size}MT"
Incoterm:                   "{incoterm}"
Vessel:                     "{vessel}"
Starting Freight Quote:     "{freight_quote}"
Broker's Demeanor:          "{broker_demeanor}"
Broker's English Level:     "{level_of_english}"
Broker's Formality:         "{formality_level}"
Verbosity:                  "{verbosity_level}"
Selected Abbreviations:     "{abbr_text}"
(Incorporate abbreviations only if they naturally fit into the conversation.)
(DWT: {dwt} and LOA: {loa}m may be mentioned naturally in the conversation)
```

{noise_injection}

Conversation Guidelines:
{conversation_mode}

2. Negotiation Flow:

- Emails should feel natural and reflect typical shipping negotiations:
  - The charterer might push back on the rate or ask for clarifications on vessel details.
  - Mention or highlight important maritime points such as "laycan," "stem," "loading window, demurrage rate per day") even if not part of the essential attributes.

3. Subject Lines & Timestamps:

- Include a concise but slightly evolving subject line for each email (e.g., "Re:", "Fwd:", minor additions, etc.).
- Add a timestamp in chronological order. Gaps between timestamps should be plausible. No mentioning of timezones.

4. Stylistic Variations:

- If formality is set to "informal" and English level is "foreign" or "intermediate," introduce mild typos or grammar slips.
- If formality is "formal" and English level is "expert," maintain a more professional and polished tone.

5. JSON Output Format:

- "email_chain": an array where each element is an object with the keys "from", "to", "subject", "timestamp", and "body".
- "labels": an object containing the following pairs. For any label in "labels" that is not applicable or not mentioned in the conversation, leave its value as an empty string:
  - "broker"
  - "commodity"
  - "load_port"
  - "discharge_port"
  - "cargo_size"
  - "incoterm"
  - "vessel"
  - "dwt"
  - "loa"
  - "starting_freight_quote_currency"
  - "starting_freight_quote"

- "final_freight_quote_currency"
- "final_freight_quote"
- "laytime_start_date"
- "laytime_end_date"
- "demurrage_currency"
- "demurrage"

6. General Guidelines:

- Aim for around {email_count} messages, but adjust as needed if more or fewer work better.
- The final freight quote must always be an actual numerical value, even if the starting freight_quote is indicated as for example mid60s.
- The chain should revolve around one cargo/commodity scenario, not multiple.
- Freight quotes may be revised or clarified by the broker, but only after an initial quote has been established and only if the charterer counters with a revised rate inquiry.
- If a noise injection is triggered, weave it naturally into one of the emails without derailing the conversation.
- The broker is called {broker_name} and the charterer is called {charterer_name}.
- Vary the currency formats as preferred (for example: USD, $, usd, us$, etc.).
- Don't recite information such as vessel names in subsequent emails if they are clear from context, unless the vessel name is revised, just mention it once.
- Don't use any markdown formatting. Just plain text.

A.2. Base-Refine (BARE)

The BARE method leverages example-driven generation through a two-stage process. Unlike AttrPrompting which uses structured attributes, BARE requires high-quality seed emails to guide generation. These seeds guide the base model to generate realistic email chains, which are then refined for coherence and consistency.

The foundation of BARE consists of 100 manually curated emails that exhibit diverse shipping scenarios, maritime terminology, negotiation dynamics, and communication styles. During base generation, the model randomly selects two emails from these seeds to guide generation of a new email chain, ensuring variety in the synthetic output. This section presents representative examples from the seed corpus, followed by the prompts used for both base generation and refinement stages.

A.2.1. Sample Seed Emails for BARE Generation

The following emails exemplify the diversity of the 100-email seed corpus used in BARE generation:

Example 1: Standard Negotiation with Maritime Terminology

Dear Mr. Alvarez,
We plan to move 6,000MT of ethanol from Busan to Davao next month. Could you share your earliest laycan dates and confirm if you offer any flexibility on demurrage?
Best regards,
Natalie

Example 2: Technical Query with Incoterms

Max,
We're looking to ship 2,700MT of specialty chemicals from Marseille to Casablanca. Could you provide a preliminary quote and confirm if you handle port dues under FOB terms?
Warmly,
Melissa

Example 3: Informal Tone with Abbreviations

Hi Lucia,
We plan to move 2,900MT of corn from Veracruz to Tampa. Could you lmk if wknd loadings incur an extra fee, and do you anticipate any dock delays?
Warmly,
Helena

Example 4: Urgent Request with Vessel Reference

Hi Ms. Taylor,
For the shipment of coal under MS Irene, we need to provision owners vessel hoses. Can you please send us the LOI and advise if any upcharge on the freight?
Thank you,
Rohan

Example 5: Brief Operational Query

Hi Ms. Vargas,
Can you please provide us with an updated ETA for MS Mangrovia?
Sincerely,
Anders

Example 6: Complex Multi-Part Query

Hello Mr. Martin,
We're looking to move 7,500MT of bitumen from Sohar to Dar es Salaam. Could you share your earliest loading window and let us know if you offer any dispatch incentives?
Regards,
Rita

Example 7: Email with Typing Errors

Dear Mr Johnson,
We have 9,100MT of sttyrene from Singapore to Manila??? Could yoy give me a ballpark rate asap?? Also, is partial load feasible???
Best,
Amin

Example 8: Non-shipping Context

Dear Mr. Lewis,
Please note that Mr. Green will leave our brokering firm at the end of the month. I will take over our contract and would be happy to meet. Any availability early next week?
Many thanks in advance and best regards,
Daniel James

A.2.2. Base Generation

The base generation stage aims to produce an initial, unpolished email conversation. It begins with a few-shot prompt to generate the first email and then iteratively appends the model's output to the prompt to generate subsequent replies, creating a continuous thread.

A.2.2.1. Starting Prompt (Few-Shot)

```
These are sample shipping emails:

Email A:
We have a 5,500MT urea shipment ready from Damietta to Mersin.
Could you advise your best freight rate and let me know
if weekend loadings are acceptable?
Thank you,
Jon
<END EMAIL A>

Email B:
Hey Mr. Dorsey,
Got a question: any chance you have space this week for 4,800MT of
bitumen from Yanbu to Jeddah? Please let me know if demurrage
might be an issue.
Cheers,
Adam
<END EMAIL B>

Now write a new email in the same style.

Email C:
```

A.2.2.2. Iterative Prompt

The following example illustrates how the iterative prompting during the first stage of BARE unfolds. Each prompt begins with one or more emails, and the model is instructed to generate the next email by appending `<Email N>`: as a placeholder for the response.

```
Email 1:
Dear Mr. Fahmy,
Can you estimate the cost of ship-borne insurance for 2,500 containers
of garments from Hong Kong to San Francisco in late December?
What's your limit on cargoes booked as ship's temperate?
Many thanks,
Feisal

<Email 2>:
```

As shown above, the first example consists of a single email (Email 1) which was generated using the prompt in Section A.2.2.1 followed by a placeholder `<Email 2>:` that induces the model to generate the second email.

```
Email 1:
Dear Mr. Fahmy,
Can you estimate the cost of ship-borne insurance for 2,500 containers
of garments from Hong Kong to San Francisco in late December?
What's your limit on cargoes booked as ship's temperate?
Many thanks,
Feisal

Email 2:
We received your enquiry concerning the shipping insurance.
We will estimate the cost and inform you as soon as possible.

<Email 3>:
```

Now the chain already contains two emails. The next placeholder (`<Email 3>:`) instructs the model to generate a third email, continuing the conversation.

```
Email 1:
Dear Mr. Fahmy,
Can you estimate the cost of ship-borne insurance for 2,500 containers
of garments from Hong Kong to San Francisco in late December?
What's your limit on cargoes booked as ship's temperate?
Many thanks,
Feisal

Email 2:
We received your enquiry concerning the shipping insurance.
We will estimate the cost and inform you as soon as possible.

Email 3:
Our underwriter has approved the insurance in the following details:
Sum insured: USD 125,000.00
Limit: USD 100,000.00
Commission: 1% of the sum insured. Our commission is USD 1,250.00.

<Email 4>:
```

Finally, the third example shows a conversation that has progressed through three emails, with the system once again prompting for the next in the chain (`<Email 4>:`).

A.2.3. Refinement

The refinement stage takes the raw, base-generated email chain and improves its quality. The model is instructed to make the conversation more coherent and realistic, correct any inconsistencies, and structure the final output into the required format.

A.2.3.1. Refinement Prompt

Please refine the following email chain to make it coherent and realistic without changing the theme, then convert it to well-formed JSON as specified.

Unrefined email chain:

{base_text}

Formatting requirements:

1. Follow the JSON structure below exactly.
2. Ensure each email in the chain is refined, logically consistent, and preserves its meaning.
3. Include a labels object at the end with as many fields as can be reasonably inferred (leave blank if uncertain).
4. Maintain the chronological order of emails. If necessary, add or estimate timestamps in the format YYYY-MM-DD HH:MM.
5. Use realistic email addresses based on names in the text.
6. Do not add content not present in the base text.

JSON structure:

```
{
  "email_chain": [
    {
      "from": "...",
      "to": "...",
      "subject": "...",
      "timestamp": "YYYY-MM-DD HH:MM",
      "body": "..."
    },
    ...
  ],
  "labels": {
```

```
    "broker": "",
    "commodity": "",
    "load_port": "",
    "discharge_port": "",
    "cargo_size": "",
    "incoterm": "",
    "vessel": "",
    "dwt": "",
    "loa": "",
    "starting_freight_quote_currency": "",
    "starting_freight_quote": "",
    "final_freight_quote_currency": "",
    "final_freight_quote": "",
    "laytime_start_date": "",
    "laytime_end_date": "",
    "demurrage_currency": "",
    "demurrage": ""
  }
}
```

A.3. Zero-Shot Prompting

Zero-shot prompting serves as a baseline to evaluate an LLMs inherent ability to generate specific content without explicit examples or detailed attributes. The prompt instructs the model on the task and desired output format but relies entirely on its pre-existing knowledge in maritime shipping.

A.3.1. Prompt

You are an expert in maritime shipping communications. Your task is to generate a synthetic email chain in JSON format that reflects a realistic negotiation between a ship broker and a charterer, covering topics such as freight rates, vessel details, laycan, and demurrage. The email chain should feel authentic and professional, but may include small talk or informal nuances.

Requirements:

1. The conversation must revolve around a single cargo shipment (e.g., crude oil, grains, etc.).
2. Provide roughly 5 to 7 emails that illustrate a typical back-and-forth negotiation.
3. Each email in the chain must contain:
   - "from": The sender's name or identifier.
   - "to": The recipient's name or identifier.
   - "subject": A concise subject line that may evolve (e.g., Re:, Fwd:, etc.).
   - "timestamp": A plausible date and time in chronological order (no specific timezone needed).
   - "body": The main text content of the email, which can include small disclaimers, shipping jargon, or mention of prior phone calls.
4. The final output must be valid JSON with exactly two top-level keys:
   - "email_chain": an array of emails (each email is an object with the properties listed above).
   - "labels": an object containing the following keys; fill in as many values as make sense based on your narrative. Any unused or irrelevant label should be left as an empty string:
     - "broker"
     - "commodity"
     - "load_port"
     - "discharge_port"
     - "cargo_size"

- "incoterm"
- "vessel"
- "dwt"
- "loa"
- "starting_freight_quote_currency"
- "starting_freight_quote"
- "final_freight_quote_currency"
- "final_freight_quote"
- "laytime_start_date"
- "laytime_end_date"
- "demurrage_currency"
- "demurrage"

5. The emails should mention a starting freight rate and potentially evolve to a final negotiated rate. Use realistic values (e.g., "low 50s," "$50.75," etc.) if they come up.
6. Avoid referencing any external tools, code, or attributes. Just perform the task as stated.

Important Notes:

- Maintain a logical and chronological flow.
- Use plausible names for the broker and charterer (e.g., "Sarah (Broker)," "John (Charterer)," etc.).
- Keep each email's content focused on the negotiation and related maritime details without clutter or repetition.
- Your answer must be only the valid JSON output with the structure described above. Do not include any extra commentary or formatting.

Please generate your response now, **only** in valid JSON format, with keys "email_chain" and "labels".

A.4. Few-Shot Prompting

Few-shot prompting represents an intermediate strategy between the highly-structured AttrPrompting and zero-shot prompting. By providing the language model with a small number of complete, high-quality examples, this method guides the generation process toward the desired style, format, and content. The model is expected to learn the underlying patterns from these examples to produce a new, similar but distinct, email chain.

A.4.1. Prompt

You are an expert in maritime shipping communications. Below are **three examples** in valid JSON format, representing how an email chain might be structured. They show the general layout and labeling but contain no real content. **Do not** reuse or paraphrase these placeholders; treat them purely as format references.

Example #1

{example_1}

Example #2

{example_2}

Example #3

{example_3}

Your Task Now:

1. Generate a completely new maritime email chain in JSON format, using the same structure as shown in the placeholders (i.e., a top-level "email_chain" array and a "labels" object).
2. Do **not** copy or paraphrase the placeholder content. Invent **new** participants, rates, ports, vessel data, and timestamps.
3. Aim for 5{7 emails, showing a realistic back-and-forth negotiation between a broker and a charterer about a single cargo shipment.
4. Include at least one mention of a starting freight rate or lumpsum offer and any subsequent changes to the rate if negotiation occurs.
5. The final output must be valid JSON, containing only:
   - "email_chain": an array of email objects (with "from", "to", "subject", "timestamp", "body").
   - "labels": an object with the specified keys and any relevant values.

6. Provide a logically consistent timeline for the emails, maintaining professional maritime negotiation details (e.g., laycan, cargo size, incoterms, vessel specs, etc.).

Output Format Reminder:
The final output must be valid JSON with exactly two top-level keys:

- "email_chain":
  an array of emails (each email is an object with the properties listed above).
- "labels":
  an object containing the following keys; fill in as many values as make sense based on your narrative. Any unused or irrelevant label should be left as an empty string:
  - "broker"
  - "commodity"
  - "load_port"
  - "discharge_port"
  - "cargo_size"
  - "incoterm"
  - "vessel"
  - "dwt"
  - "loa"
  - "starting_freight_quote_currency"
  - "starting_freight_quote"
  - "final_freight_quote_currency"
  - "final_freight_quote"
  - "laytime_start_date"
  - "laytime_end_date"
  - "demurrage_currency"
  - "demurrage"

A.5. API Interaction Costs

The extensive use of LLMs for synthetic data generation incurs direct financial costs, which are an important practical consideration for research and application. This section provides a transparent summary of the total expenditure for interacting with the various commercial APIs used in the experiments described in this thesis. Table A.1 itemizes these costs by model.

Model	**Cost (USD)**
Mistral	27.11
Gemini	0.00
GPT4	77.06
Claude	49.18
DeepSeek	4.78
Total	**158.13**

Table A.1.: API interaction costs for synthetic email generation experiments conducted in this thesis. Costs are reported in USD and reflect the total expenditure for all generation methods and evaluation tasks. Gemini incurred no costs due to free tier usage during the experimental period.

A.6. Maritime Entity Annotation Prompt

The following prompt was used to annotate Wikipedia and news articles for maritime entity extraction training data:

```
You are an expert in maritime and chartering terminology.  Your task is to
identify and extract named entities from multiple texts.

REQUIRED LABELS (must identify all occurrences):
- location: cities, countries, ports, but non-generic (e.g., 'Port of Rotterdam'
= location but 'port' != location)
- vessel: vessel names (e.g., 'MTM Rotterdam' = vessel but 'container ship' !=
vessel)
- incoterm: trade terms such as FOB, CIF, CFR, EXW, DDP, etc.
- commodity: types of cargo such as gold, corn, wheat, metals, crude oil, coal,
etc.

For each text, if suitable, select up to 5 additional labels that are
most relevant.

TEXTS TO ANNOTATE:
[TEXT 1]
{text}

INSTRUCTIONS:
1. For each text, identify all entities that match the required labels
2. Add up to 5 additional entity types per text if they are prominently featured
3. Return ONLY an array of arrays in JSON format
4. Each entity should be in the format: "entity_text <> entity_type"
5. The outer array should have one inner array for each text, in the same order

OUTPUT FORMAT:
[
["Port of Rotterdam <> location", "FOB <> incoterm"],
["MTM Singapore <> vessel", "crude oil <> commodity"],
...
]

Return ONLY the JSON array of arrays with the extracted entities.
```

B. DetectGPT

In this chapter we provide the visual results for the DetectGPT authorship analysis, which is used to assess how closely the synthetic emails resemble human-written text. As outlined in the methodology (Chapter 4), DetectGPT distinguishes between human and machine-generated content by measuring text likelihood under perturbation. The following figure visualizes the score distributions for each data generation method compared against the human-authored ENRON baseline, complementing the analysis presented in Section 5.1.6.

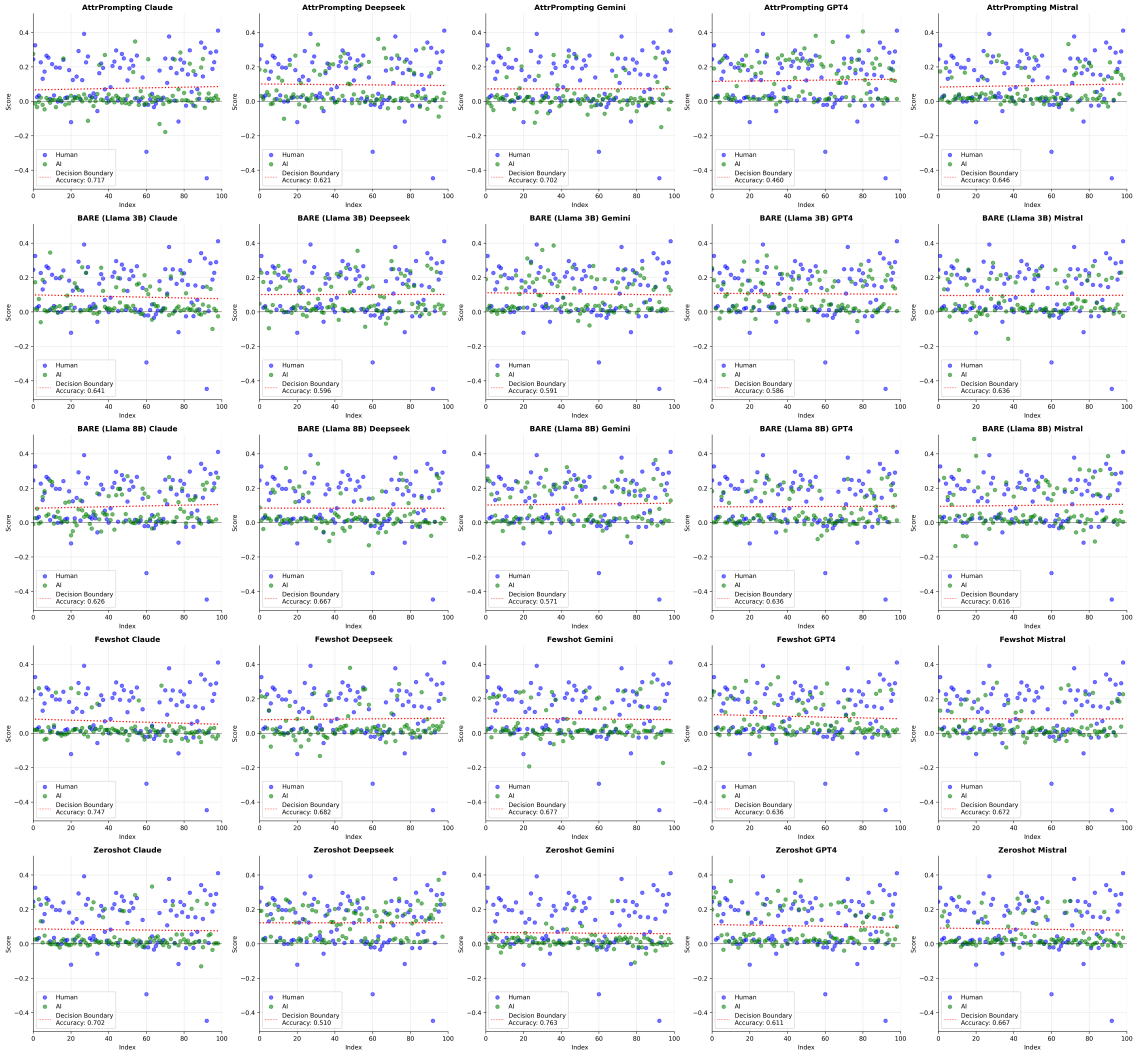

Figure B.1.: DetectGPT Authorship Detection

C. Text Embeddings

The following t-SNE visualizations provide comprehensive views of entity embeddings across all seven datasets from the out-of-distribution NER benchmark, complementing the analysis presented in Section 5.2.1.1. Each visualization compares the baseline DeBERTa-v3-small model with contrastively fine-tuned variants using BGE and SBERT similarity targets, demonstrating the consistent improvements in entity clustering quality achieved through contrastive pretraining.

Figure C.1.: t-SNE visualization of entity embeddings for the *Literature* dataset from CrossNER (Liu et al., 2021b), comparing baseline DeBERTa-v3-small with contrastively fine-tuned variants.

Figure C.2.: t-SNE visualization of entity embeddings for the *Movie* dataset from MIT (Liu et al., 2013), comparing baseline DeBERTa-v3-small with contrastively fine-tuned variants.

Figure C.3.: t-SNE visualization of entity embeddings for the *Music* dataset from CrossNER (Liu et al., 2021b), comparing baseline DeBERTa-v3-small with contrastively fine-tuned variants.

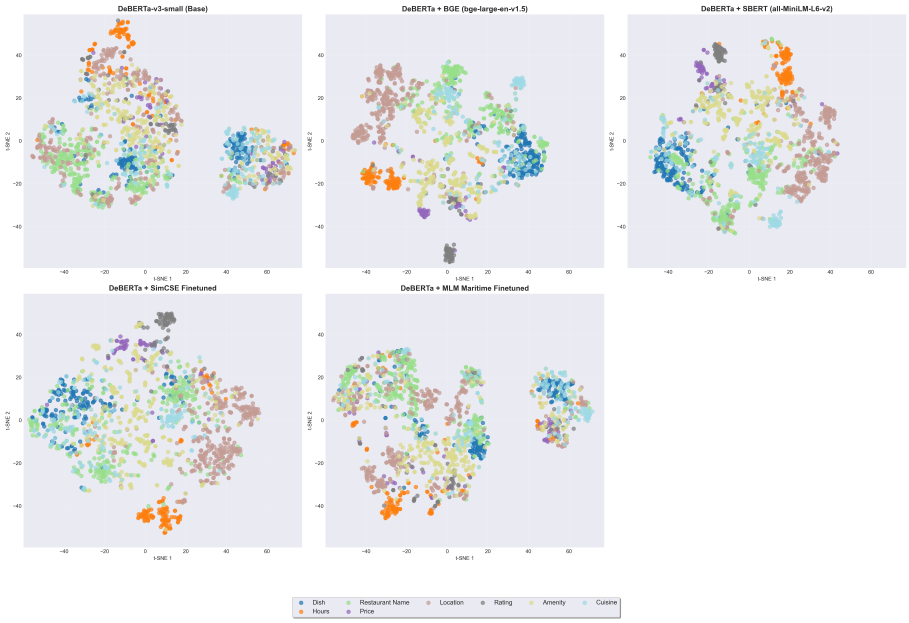

Figure C.4.: t-SNE visualization of entity embeddings for the *Restaurant* dataset from MIT (Liu et al., 2013), comparing baseline DeBERTa-v3-small with contrastively fine-tuned variants.

Figure C.5.: t-SNE visualization of entity embeddings for the *Science* dataset from CrossNER (Liu et al., 2021b), comparing baseline DeBERTa-v3-small with contrastively fine-tuned variants.

D. OOD Evaluation

Detailed entity-level F1 scores for OOD NER evaluation are presented in the following tables, offering granular insights into how different GLiNER encoder variants perform on specific entity types within each domain. These breakdowns complement the aggregate results presented in Table 5.2.1.2 of the main text and reveal entity-specific strengths and weaknesses of different pretraining strategies.

The detailed results confirm several key patterns observed in the main analysis. First, contrastive pretraining methods (BGE, SBERT, SimCSE) consistently outperform the baseline on entities requiring semantic disambiguation, such as distinguishing *person* from *scientist* in the Science domain or *person* from *politician* in the Politics domain. Second, the MLM approach shows unexpected strengths on concrete, location-based entities due to geographical pattern transfer from the maritime training corpus. Finally, the baseline model retains advantages on abstract, functionally-defined entities where contrastive constraints may hinder flexible recognition.

Entity	**GLiNER Base**	**MLM**	**SimCSE**	**BGE**	**SBERT**
award	**64.85%**	53.85%	62.50%	57.55%	55.63%
book	**79.09%**	66.74%	72.48%	70.75%	73.08%
country	34.97%	67.86%	69.61%	73.12%	**73.30%**
event	**44.87%**	25.64%	33.51%	34.48%	32.56%
literary genre	32.02%	42.16%	41.87%	41.76%	**42.36%**
location	54.61%	57.61%	**62.66%**	60.34%	61.47%
magazine	**88.50%**	69.86%	86.21%	83.20%	85.25%
organization	**64.26%**	50.83%	57.48%	58.27%	59.06%
person	47.57%	46.95%	51.09%	**51.82%**	49.04%
poem	**47.92%**	35.52%	44.92%	42.11%	41.44%
writer	67.54%	76.75%	77.83%	79.53%	**80.11%**
OVERALL	60.82%	58.12%	63.47%	63.38%	**64.06%**

Table D.1.: Entity-level F1 Scores for Literature Dataset

Entity	GLiNER Base	MLM	SimCSE	BGE	SBERT
actor	**80.72%**	58.29%	67.22%	58.14%	59.13%
average ratings	7.64%	**29.97%**	14.36%	16.70%	19.03%
character	**18.28%**	10.04%	11.85%	9.90%	10.00%
director	**80.29%**	64.60%	69.79%	64.48%	62.99%
genre	36.95%	33.59%	36.38%	**38.31%**	37.50%
plot	25.72%	25.29%	27.22%	**29.66%**	28.23%
rating	33.86%	37.87%	**44.42%**	42.06%	38.14%
review	**6.10%**	2.51%	2.39%	2.78%	1.11%
song	**36.22%**	12.90%	17.24%	19.30%	15.79%
title	**12.89%**	11.12%	10.19%	11.20%	8.35%
trailer	13.16%	46.30%	63.29%	65.75%	**73.53%**
year	63.08%	**71.59%**	71.01%	70.45%	69.09%
OVERALL	**43.48%**	37.27%	41.49%	40.31%	38.77%

Table D.2.: Entity-level F1 Scores for Movie Dataset

Entity	GLiNER Base	MLM	SimCSE	BGE	SBERT
country	58.20%	**60.76%**	58.42%	59.34%	60.66%
election	**72.01%**	62.87%	65.70%	57.38%	63.49%
event	52.37%	45.35%	48.58%	52.00%	**55.15%**
location	76.08%	78.70%	79.32%	80.03%	**81.61%**
organization	62.81%	58.70%	64.05%	64.66%	**65.58%**
person	78.52%	75.62%	81.59%	79.51%	**81.90%**
political party	**68.82%**	58.77%	60.40%	62.46%	62.92%
politician	**70.53%**	65.32%	69.11%	69.27%	70.52%
OVERALL	**68.65%**	64.03%	66.73%	66.54%	68.31%

Table D.3.: Entity-level F1 Scores for Politics Dataset

Entity	GLiNER Base	MLM	SimCSE	BGE	SBERT
academic journal	**87.19%**	70.99%	79.30%	80.58%	74.69%
astronomical object	**76.53%**	56.69%	60.61%	65.58%	60.71%
award	**55.24%**	44.03%	48.06%	45.05%	45.99%
chemical compound	**63.48%**	44.63%	47.33%	49.62%	49.41%
chemical element	**71.89%**	52.53%	53.11%	60.94%	59.32%
country	41.86%	42.99%	48.19%	**51.11%**	49.48%
discipline	**50.98%**	32.35%	38.27%	37.59%	33.03%
enzyme	**61.33%**	42.93%	47.50%	54.09%	53.01%
event	**27.21%**	10.63%	12.90%	14.61%	14.35%
location	**61.75%**	53.98%	60.61%	61.14%	56.28%
organization	**77.96%**	58.71%	66.79%	65.97%	60.88%
person	45.69%	55.51%	67.12%	66.00%	**67.41%**
protein	**50.13%**	43.97%	40.44%	45.76%	46.88%
scientist	10.15%	69.06%	83.62%	81.71%	**85.59%**
theory	**22.61%**	9.41%	10.58%	11.83%	11.46%
university	**87.22%**	76.43%	81.85%	80.92%	80.47%
OVERALL	57.89%	51.65%	59.33%	**60.55%**	59.00%

Table D.4.: Entity-level F1 Scores for Science Dataset

E. GLiNER

This appendix chapter provides comprehensive visualizations of GLiNER performance across different entity types and experimental conditions that complement the main results presented in Section 5.2.1. The detailed breakdowns reveal entity-specific challenges and confirm the patterns observed in the aggregate analysis.

E.1. Entity-Specific Performance Analysis

The following heatmaps present F1 scores for individual maritime entity types, revealing distinct extraction challenges across different generation methods and model configurations. Figure E.1 details the results for vessel extraction, Figure E.2 for ports, and Figure E.3 for commodities, with each map illustrating the impact of different model configurations.

E.2. Performance Distribution Analysis

Additionally, the violin plots shown in Figure E.4 provide detailed views of performance distributions across different experimental factors, revealing variance patterns and outliers not apparent in aggregate metrics.

These detailed visualizations confirm the main findings presented in the results section while revealing additional nuances in GLiNER's performance characteristics. The entity-specific heatmaps demonstrate that extraction difficulty varies substantially by entity type, with commodity extraction being most robust and incoterm extraction presenting the greatest challenges. The distribution plots reveal that finetuning strategy has a more profound impact on performance consistency than encoder pretraining method, supporting the conclusion that domain-specific adaptation is critical for maritime entity extraction tasks.

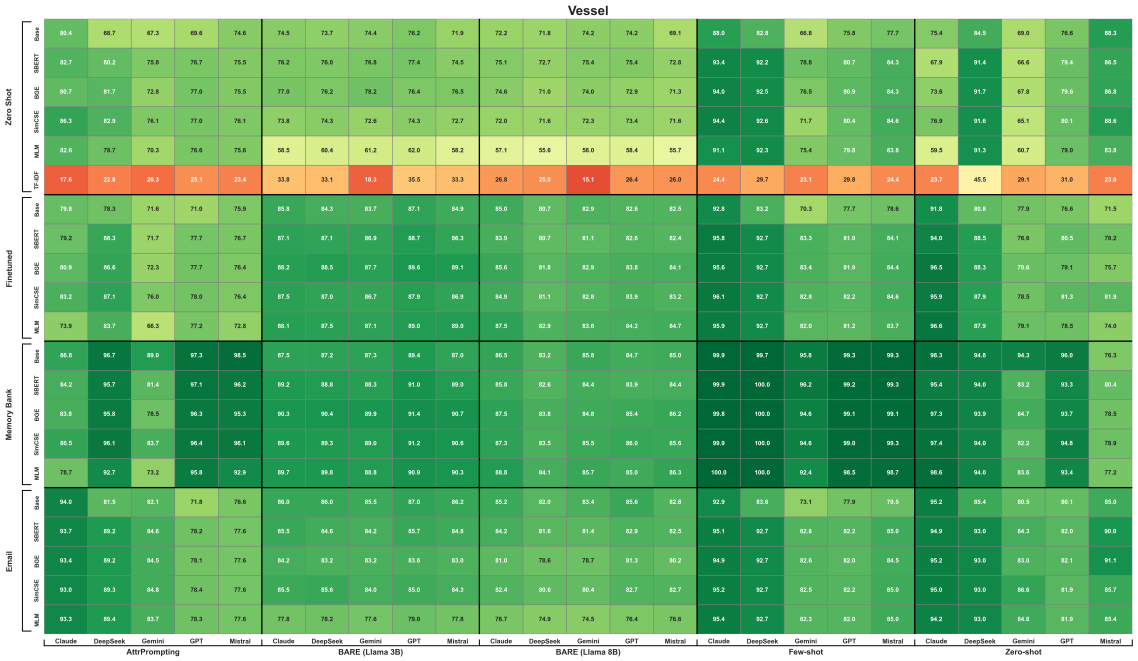

Figure E.1.: Heatmap of F1 scores for vessel entity extraction across all GLiNER variants and synthetic email generation configurations. Rows represent different GLiNER models grouped by finetuning strategy (Zero Shot, Finetuned, Memory Bank, Email) and encoder pretraining method. Columns show combinations of generation methods and generation models. Cell values indicate macro-averaged F1 scores as percentages, with color coding from red (poor performance) to green (excellent performance). The results show that vessel extraction is particularly challenging for zero-shot models on BARE generations, while memory bank and email-specific finetuning achieve consistently high performance across all generation methods. Notable patterns include the difficulty of extracting vessels from Claude-generated BARE emails and the superior performance on structured generation methods like attribute prompting.

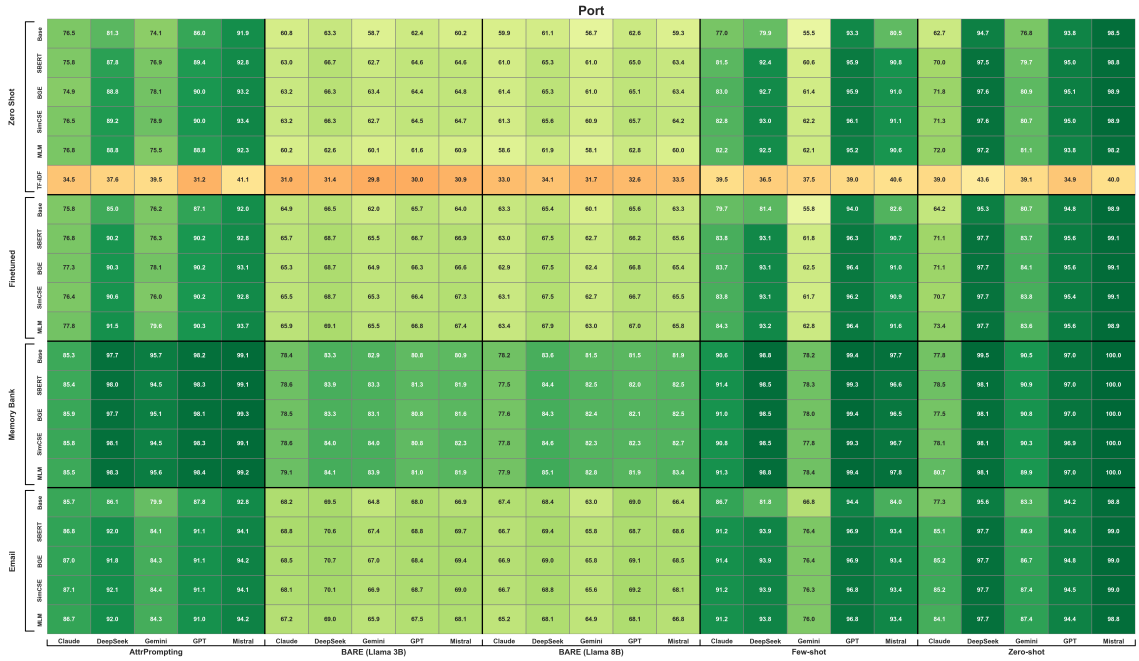

Figure E.2.: Heatmap of F1 scores for port entity extraction across all GLiNER variants and synthetic email generation configurations. The layout follows the same structure as Figure E.1. Port extraction demonstrates more consistent performance compared to vessel extraction, with fewer extreme variations across generation methods. Zero-shot models achieve moderate success on structured generations but still struggle with BARE-generated content. The memory bank approach shows particular strength in disambiguating ports from vessels, a critical challenge in maritime entity recognition where many location names can refer to both geographical locations and vessel names.

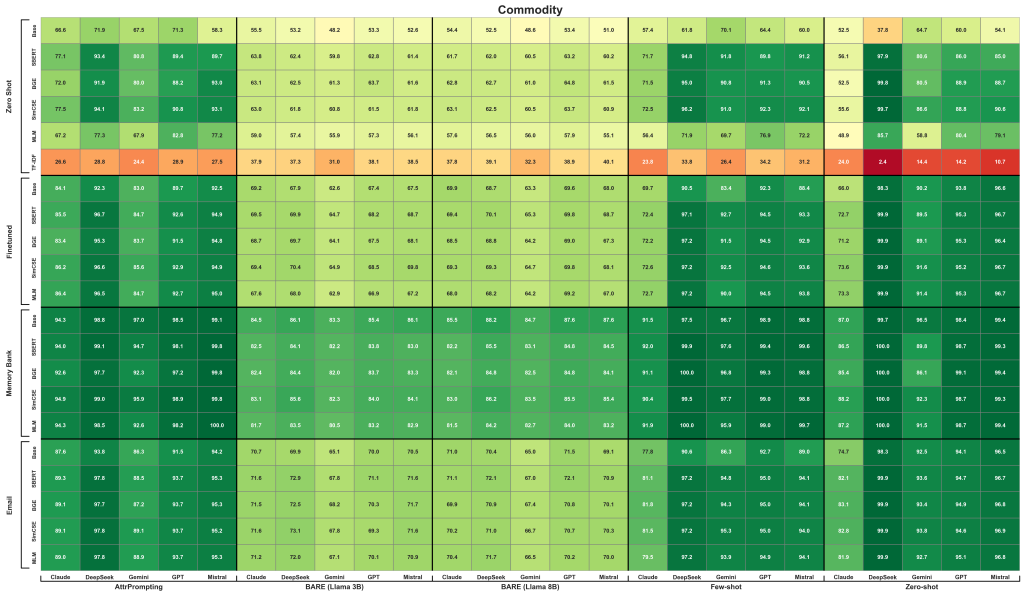

Figure E.3.: Heatmap of F1 scores for commodity entity extraction across all GLiNER variants and synthetic email generation configurations. Commodity extraction represents the most successful entity type across all model variants, with consistently high F1 scores (predominantly green cells) even in zero-shot settings. This success can be attributed to the concrete nature of commodity terms (e.g., "crude oil," "iron ore," "wheat") which have clear semantic boundaries and limited ambiguity compared to vessel names or technical abbreviations. The uniformly high performance across generation methods suggests that commodity terms are robustly represented in the GLiNER training data and maintain consistent linguistic patterns across different synthetic generation approaches.

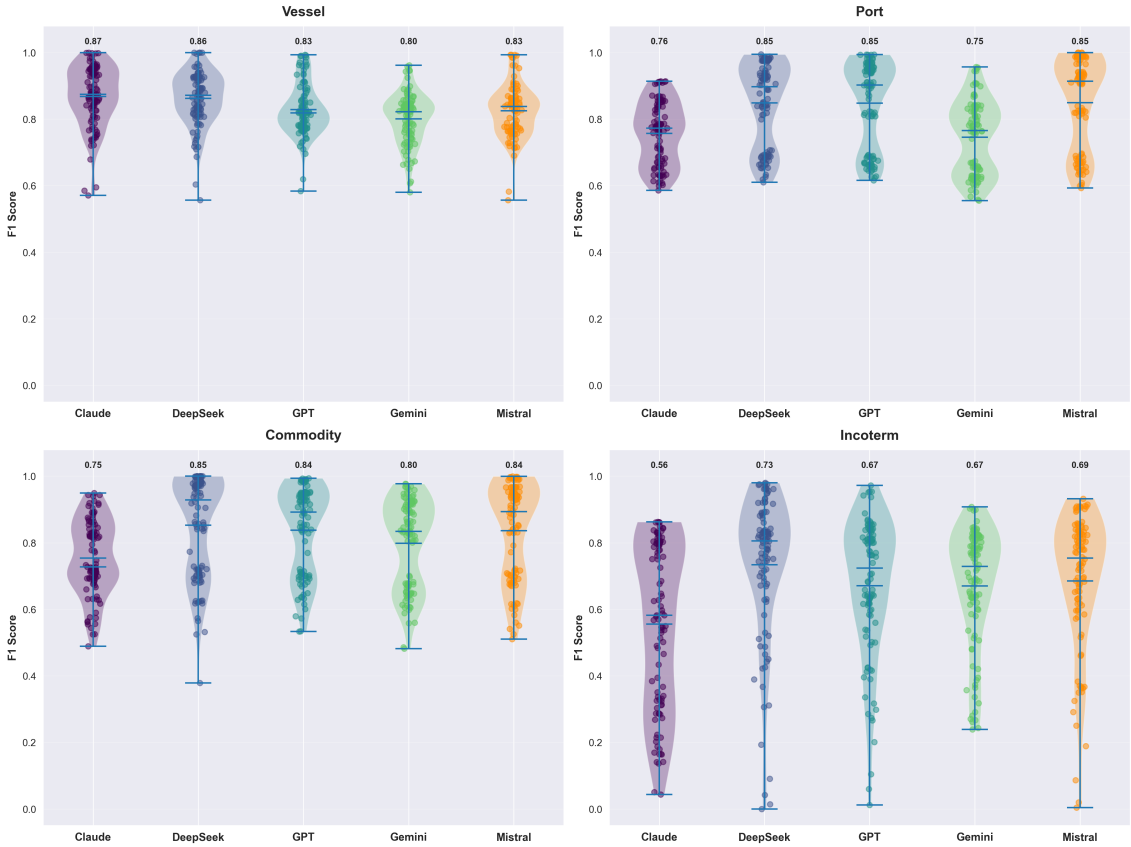

Figure E.4.: Violin plots showing F1 score distributions for each of the four maritime entity types: vessel, port, commodity, and incoterm. Each violin shows the full distribution of F1 scores across all GLiNER model variants and generation configurations, with individual data points overlaid as dots. The width of each violin indicates the density of scores at that performance level. Commodity extraction shows the highest and most consistent performance with scores concentrated above 0.8, while incoterm extraction exhibits the most variable and generally lowest performance. Port and vessel extraction show intermediate performance with notable variance, reflecting the challenges of disambiguating these semantically similar entity types. The bimodal distribution visible in some entity types indicates distinct performance regimes corresponding to different finetuning strategies.

F. CodeIE vs. Standard LLM Performance Analysis

This section provides the detailed results visualized in section 5.2.2, comparing code-based and standard LLMs in information extraction performance. Models are arranged according to their F1 macro score in descending order.

Model	Type	Size	F1 Macro	Precision	Recall	Avg. Time (s)
Qwen2.5-1.5B	Standard	1.5B	0.880	0.880	0.880	4.38
StableLM-3B	Standard	3B	0.874	0.883	0.865	9.67
OpenLlama-3B	Standard	3B	0.827	0.832	0.820	11.80
StableLM-2-1.6B	Standard	1.6B	0.734	0.777	0.695	5.48
TinyLlama-1.1B	Standard	1.1B	0.725	0.749	0.700	4.20
Qwen2.5-0.5B	Standard	0.5B	0.708	0.724	0.695	3.00
Qwen2.5-Coder-3B	Code	3B	0.686	0.784	0.635	11.34
Phi-2 (2.7B)	Standard	2.7B	0.577	0.708	0.485	9.09
Phi-1.5 (1.3B)	Standard	1.3B	0.409	0.566	0.385	5.20
CodeGen-2B-NL	Code	2.7B	0.326	0.406	0.280	10.32
CodeGen-2B	Code	2B	0.203	0.230	0.185	10.31
CodeGen-2B	Code	2.7B	0.190	0.218	0.170	10.13
Phi-1 (350M)	Standard	350M	0.171	0.195	0.165	5.20
Flan-T5-Large (780M)	Standard	780M	0.102	0.929	0.065	1.96
CodeGen-350M	Code	350M	0.071	0.072	0.070	6.10
CodeGen-350M-Mono	Code	350M	0.049	0.053	0.045	6.14
Flan-T5-Small (80M)	Standard	80M	0.032	0.286	0.020	1.35
StarCoder2-3B	Code	3B	0.016	0.034	0.010	11.25
CodeParrot-Small (110M)	Code	110M	0.010	0.333	0.005	2.08
CodeT5-Base (220M)	Code	220M	0.007	0.008	0.005	1.99
Pythia-160M	Standard	160M	0.000	0.000	0.000	2.16
TinyStarCoder (164M)	Code	164M	0.000	0.000	0.000	2.47
CodeT5-Small (60M)	Code	60M	0.000	0.000	0.000	1.14
CodeGPT-Java (1.4B)	Code	1.4B	0.000	0.000	0.000	2.03

Table F.1.: Performance comparison of code-specialized and standard language models on the maritime entity extraction task. Models are ranked by F1 Macro score, with model type (Code or Standard), parameter count, overall performance metrics (F1 Macro, Precision, and Recall), and average inference time per sample shown. Standard models generally outperform code-specialized models, with the top 6 performers being standard models. The best performing model (Qwen2.5-1.5B) achieves an F1 Macro of 0.880 with an inference time of 4.38 seconds. Smaller models like Flan-T5 variants show faster inference times (1.14-1.96s) but significantly lower performance. Many code models fail to extract entities correctly despite longer inference times.

G. Qwen-2.5-0.5B: Template Filling

Finally, in this appendix chapter, we present additional evaluation results for the Qwen-2.5-0.5B model variants discussed in Chapter 5.2.3. While the main text focused on vessel and Incoterms extraction to illustrate key performance patterns, this section provides complete results across all four entity types for transparency and reproducibility.

G.1. Entity-Specific Performance Heatmaps

Figures G.1 (Commodity) and G.2 (Port) complement the vessel and Incoterm heatmaps presented in the main text. These visualizations confirm the difficulties introduced by BARE generations, but show consistent, high-performance metrics.

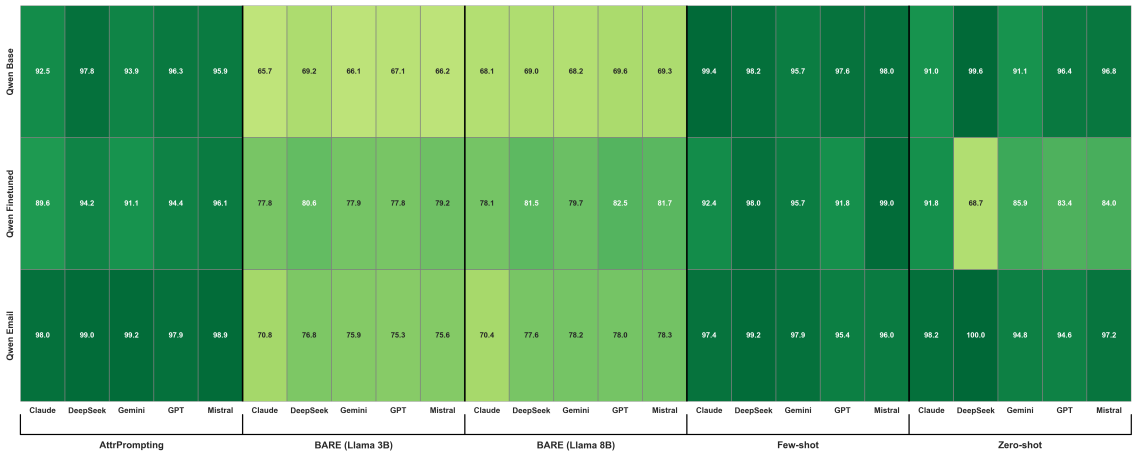

Figure G.1.: Extraction Heatmap Qwen Commodity

G.1.1. Entity-Specific Performance by Generation Model

Figure G.3 dissects the overall performance shown in Section 5.2.3 to reveal entity-specific patterns across generation models. While commodity and port extraction

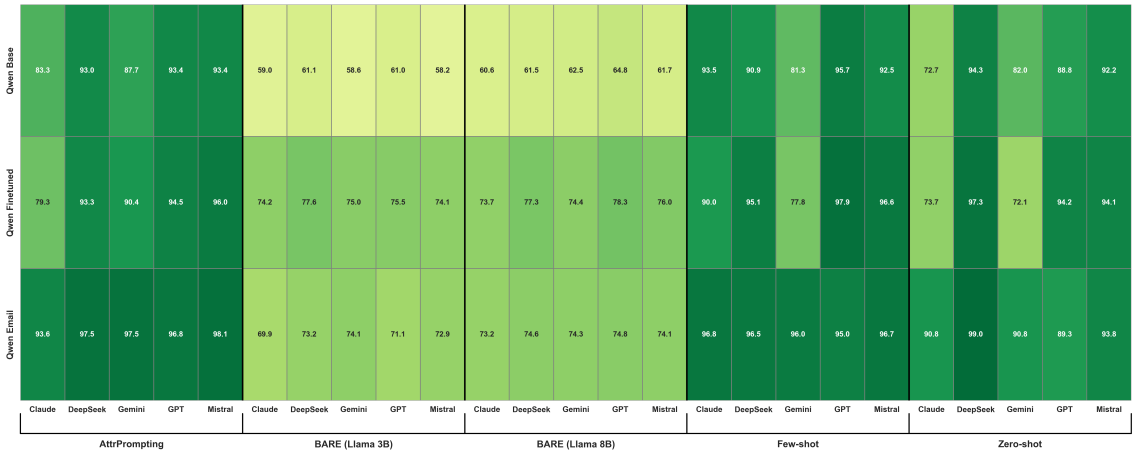

Figure G.2.: Extraction Heatmap Qwen Port

show relatively consistent performance across all generation models (narrow distributions with means above 0.80), vessel and Incoterms extraction exhibit greater variability, particularly for Claude-generated content.

G.1.2. Overall Performance Distributions

Figure G.4 presents three complementary views of the overall extraction performance, aggregating across all entity types to highlight systematic patterns in the evaluation results.

G.2. Detailed BARE Generation Results

Table G.2 provides micro-averaged precision, recall, and F1 scores for all BARE-generated datasets. These metrics offer more profound insight into the precision-recall trade-offs discussed in Section 5.2.3.1. The table format allows direct comparison of how each model variant handles entity absence across different generation models and base model sizes (Llama 3B vs. 8B).

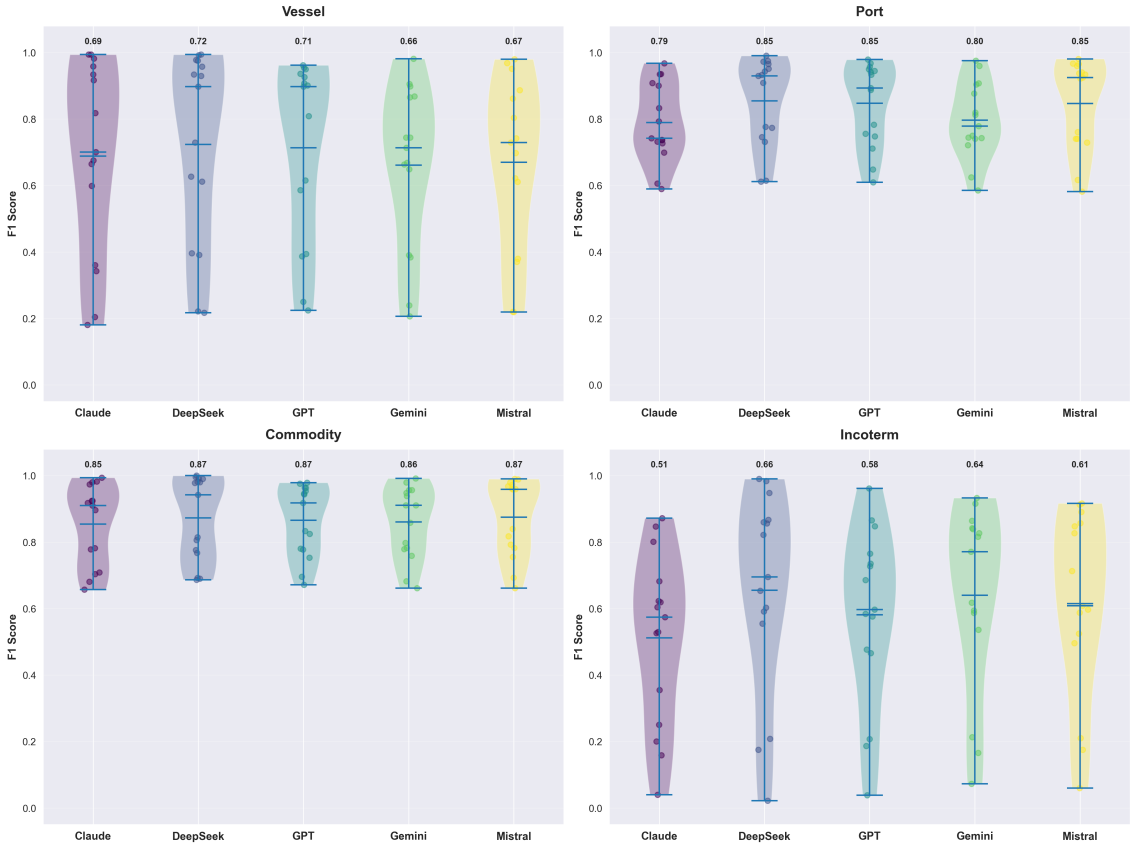

Figure G.3.: Entity-specific extraction performance distributions across generation models. Each violin plot shows the distribution of F1 scores aggregated across all Qwen model variants and generation methods. Dots represent individual evaluation results, with horizontal lines indicating median and mean values (shown numerically above each plot). The wider sections of each violin indicate more common score ranges, revealing that vessel and Incoterms extraction suffer from bimodal distributions, particularly pronounced in Claude-generated emails.

Model	Generation	Vessel			Commodity			Incoterm			Port		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
BARE Llama 3B													
Qwen Base													
	Claude	.074	.512	.129	.595	.673	.631	.042	.223	.071	.615	.652	.633
	DeepSeek	.071	.415	.121	.577	.611	.594	.043	.206	.071	.695	.564	.622
	Gemini	.073	.460	.126	.549	.562	.556	.037	.187	.062	.662	.548	.599
	GPT	.087	.529	.149	.591	.642	.616	.053	.264	.089	.668	.619	.643
	Mistral	.075	.543	.132	.580	.609	.594	.039	.194	.065	.664	.562	.609
Qwen Email													
	Claude	.101	.583	.172	.621	.724	.668	.188	.536	.278	.700	.839	.763
	DeepSeek	.102	.476	.167	.616	.695	.653	.193	.461	.273	.791	.743	.766
	Gemini	.103	.525	.172	.618	.680	.648	.161	.415	.232	.766	.730	.748
	GPT	.112	.557	.186	.621	.720	.667	.196	.529	.286	.741	.795	.767
	Mistral	.098	.586	.168	.607	.696	.649	.195	.524	.284	.769	.767	.768
Qwen Finetuned													
	Claude	.161	.465	.239	.651	.717	.682	.354	.307	.329	.609	.852	.710
	DeepSeek	.123	.333	.180	.657	.660	.658	.305	.256	.278	.730	.796	.762
	Gemini	.142	.343	.201	.659	.638	.648	.294	.246	.268	.695	.758	.725
	GPT	.140	.421	.211	.661	.687	.674	.324	.276	.298	.674	.828	.743
	Mistral	.113	.362	.172	.656	.658	.657	.259	.224	.240	.677	.790	.729
BARE Llama 8B													
Qwen Base													
	Claude	.080	.435	.136	.610	.668	.638	.059	.280	.098	.635	.649	.642
	DeepSeek	.082	.337	.131	.594	.608	.601	.053	.235	.087	.726	.572	.640
	Gemini	.073	.332	.120	.585	.569	.577	.049	.198	.078	.694	.561	.620
	GPT	.096	.424	.156	.613	.647	.630	.065	.290	.106	.710	.641	.674
	Mistral	.077	.400	.129	.624	.649	.636	.057	.257	.093	.696	.587	.637
Qwen Email													
	Claude	.119	.534	.195	.617	.722	.665	.212	.505	.298	.722	.851	.781
	DeepSeek	.127	.428	.196	.626	.697	.659	.186	.429	.260	.794	.739	.766
	Gemini	.096	.364	.152	.617	.649	.632	.200	.411	.269	.774	.724	.748
	GPT	.112	.403	.175	.643	.726	.682	.197	.466	.277	.779	.793	.786
	Mistral	.107	.456	.173	.649	.729	.687	.206	.487	.290	.775	.771	.773
Qwen Finetuned													
	Claude	.185	.391	.251	.675	.721	.697	.310	.285	.297	.605	.855	.709
	DeepSeek	.149	.269	.192	.653	.655	.654	.313	.260	.284	.735	.801	.766
	Gemini	.139	.223	.172	.677	.640	.658	.303	.242	.269	.678	.774	.723
	GPT	.150	.335	.207	.669	.691	.680	.333	.290	.310	.689	.835	.755
	Mistral	.122	.281	.170	.692	.699	.696	.294	.241	.265	.687	.804	.741

Table G.1.: Qwen model performance on BARE-generated emails: Micro-averaged metrics

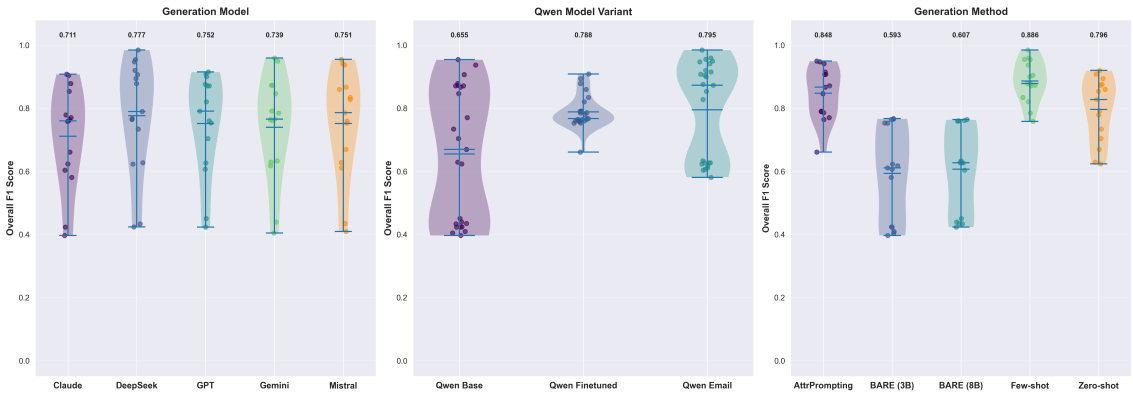

Figure G.4.: Overall extraction performance distributions across three analytical dimensions. Left: Performance by generation model confirms DeepSeek as the most consistent generator (mean 0.777, narrow distribution) while Claude produces the most challenging content (mean 0.711, wide distribution). Center: Qwen model variants exhibit clear performance stratification, with Email finetuning achieving the highest scores (0.798). In contrast, Finetuned shows bimodality due to its BARE specialization. Right: Generation methods reveal the expected performance gradient, with BARE methods showing both lower means and higher variance compared to other approaches.