



School of Science and Technology
MSc in Information Systems

Master Thesis

Multimodal Entity Resolution for Retail Receipt Line Items Using Semi-Structured Text and Images

Maria Drinia

Supervisor: Dimitrios Karapiperis

Patras, Greece, 2026

Theses / Dissertations remain the intellectual property of students (“authors/creators”), but in the context of open access policy they grant to the HOU a non-exclusive license to use the right of reproduction, customisation, public lending, presentation to an audience and digital dissemination thereof internationally, in electronic form and by any means for teaching and research purposes, for no fee and throughout the duration of intellectual property rights. Free access to the full text for studying and reading does not in any way mean that the author/creator shall allocate his/her intellectual property rights, nor shall he/she allow the reproduction, republication, copy, storage, sale, commercial use, transmission, distribution, publication, execution, downloading, uploading, translating, modifying in any way, of any part or summary of the dissertation, without the explicit prior written consent of the author/creator. Creators retain all their moral and property rights.

Multimodal Entity Resolution for Retail Receipt Line Items Using Semi-Structured Text and Images

Maria Drinia

Supervising Committee

Dimitrios Karapiperis, [Affiliation]

[Committee Member 2], [Affiliation]

[Committee Member 3], [Affiliation]

Patras, Greece, 2026

Table of Contents

Abstract	1
Abstract (Περίληψη στα Ελληνικά)	2
Acknowledgements	3
List of Tables	3
List of Figures	4
1 Chapter 1: Introduction	5
1.1 1.1 Background and Motivation	5
1.2 1.2 Research Problem	7
1.3 1.3 Research Questions	7
1.4 1.4 Approach Overview	7
1.5 1.5 Contributions	8
1.6 1.6 Thesis Structure	8
2 Chapter 2: Theoretical Background	9
2.1 2.1 Entity Resolution	9
2.1.1 2.1.1 Problem Definition	9
2.1.2 2.1.2 Traditional Approaches	10
2.1.3 2.1.3 Neural Entity Resolution	11
2.2 2.2 Transformer Architecture	11
2.2.1 2.2.1 Self-Attention Mechanism	11
2.2.2 2.2.2 BERT and Pre-trained Language Models	12
2.2.3 2.2.3 Efficient Transformers and Knowledge Distillation	14
2.3 2.3 Vision Transformers and Document Understanding	14
2.3.1 2.3.1 Vision Transformer (ViT)	14
2.3.2 2.3.2 Document Image Transformer (DiT)	14
2.3.3 2.3.3 LayoutLMv3	15
2.4 2.4 Contrastive Learning	15
2.4.1 2.4.1 Contrastive Learning Framework	15
2.4.2 2.4.2 InfoNCE Loss	15
2.4.3 2.4.3 Hard Negative Mining	16
2.5 2.5 Multimodal Fusion Strategies	17
2.5.1 2.5.1 Early Fusion	17
2.5.2 2.5.2 Late Fusion	17
2.5.3 2.5.3 Hybrid Fusion (Cross-Attention)	17
2.6 2.6 Approximate Nearest Neighbor Search	18
3 Chapter 3: Related Work	18
3.1 3.1 Entity Resolution with Deep Learning	18
3.2 3.2 Multimodal Entity Resolution	19
3.3 3.3 Receipt and Document Understanding	20
3.4 3.4 Contrastive Learning for Retrieval	21
3.5 3.5 Greek Natural Language Processing	21
3.6 3.6 Product Matching in E-commerce	22
3.7 3.7 Summary and Research Gap	22
3.8 3.8 Recent Developments: LLMs for Entity Resolution	23

4	Chapter 4: Data Description	24
4.1	4.1 Data Source	24
4.2	4.2 Scope and Selection Criteria	25
4.3	4.3 Dataset Components	25
4.3.1	4.3.1 Receipt Line Items	25
4.3.2	4.3.2 Product Catalog	26
4.3.3	4.3.3 Ground Truth	26
4.3.4	4.3.4 Receipt Images	27
4.4	4.4 Dataset Statistics	28
4.4.1	4.3.5 Dataset Examples	29
4.5	4.5 Data Anonymization	42
4.6	4.6 Data Preprocessing	43
4.6.1	4.6.1 Greek Text Normalization	43
4.6.2	4.6.2 Image Preprocessing	45
4.7	4.7 Data Partitioning	47
5	Chapter 5: Methodology	47
5.1	5.1 Problem Formulation	47
5.2	5.2 Overall Architecture	48
5.3	5.3 Experiment Matrix	49
5.4	5.4 Text Encoders	50
5.4.1	5.4.1 TF-IDF Baseline (B1)	50
5.4.2	5.4.2 BM25 Baseline (B2)	50
5.4.3	5.4.3 GreekBERT (T1)	50
5.4.4	5.4.4 mMiniLM (T2)	51
5.4.5	5.4.5 OpenAI text-embedding-3-small (B3)	51
5.5	5.5 Image Encoder	52
5.5.1	5.5.1 DiT (I1, M1-M3)	52
5.6	5.6 Multimodal Fusion	52
5.6.1	5.6.1 Early Fusion (M1)	52
5.6.2	5.6.2 Late Fusion (M2)	52
5.6.3	5.6.3 Hybrid Fusion with Cross-Attention (M3)	53
5.6.4	5.6.4 LayoutLMv3 (L1)	54
5.7	5.7 Training Procedure	54
5.7.1	5.7.1 Pair Generation	54
5.7.2	5.7.2 Loss Function	56
5.7.3	5.7.3 Staged Training	56
5.7.4	5.7.4 Optimization	56
5.7.5	5.7.5 Data Augmentation	57
5.8	5.8 Evaluation Protocol	57
5.8.1	5.8.1 Retrieval Setup	57
5.8.2	5.8.2 Metrics	57
6	Chapter 6: Experimental Setup	58
6.1	6.1 Implementation	58
6.2	6.2 Pre-trained Models	59
6.3	6.3 Hyperparameters	59
6.4	6.4 Hardware and Computing Environment	60

6.5	6.5 Reproducibility	61
6.6	6.6 Evaluation Procedure	62
6.7	6.7 Step-by-Step Experiment Procedure	62
6.7.1	6.7.1 Statistical Baselines (B1, B2)	62
6.7.2	6.7.2 Commercial API Baseline (B3)	63
6.7.3	6.7.3 Neural Text Models (T1, T2)	63
6.7.4	6.7.4 What the System Learns from Training Data	64
6.7.5	6.7.5 Image Probe	67
6.7.6	6.7.6 Ablation Studies	67
7	Chapter 7: Experimental Results	68
7.1	7.1 Main Results	68
7.1.1	7.1.1 Baseline Performance	70
7.1.2	7.1.2 Neural Text Models vs. Baselines	70
7.1.3	7.1.3 Image Probe: Does Visual Information Help?	71
7.1.4	7.1.4 Summary of Model Comparison	73
7.2	7.2 End-to-End Evaluation: From Receipt to Product Match	74
7.2.1	7.2.1 Easy Cases: Both Models Succeed	74
7.2.2	7.2.2 Medium Cases: Neural Model Succeeds, Baseline Fails	75
7.2.3	7.2.3 Very Hard Cases: All Models Struggle	76
7.2.4	7.2.4 Summary of End-to-End Observations	77
7.3	7.3 Detailed Retrieval Metrics	78
7.3.1	7.3.1 Key Observations	78
7.4	7.4 Training Dynamics	79
7.5	7.5 Ablation Studies	83
7.5.1	7.5.1 Embedding Dimension	83
7.5.2	7.5.2 Temperature Parameter	83
7.5.3	7.5.3 Late Fusion Weight	85
7.6	7.6 Per-Category Analysis	85
7.7	7.7 OCR Quality Impact	86
7.8	7.8 Error Analysis	89
7.8.1	7.8.1 Error Categories	89
7.8.2	7.8.2 Qualitative Examples	90
7.8.3	7.8.3 Visual Information Analysis	90
7.9	7.9 Summary of Results	91
8	Chapter 8: Discussion	91
8.1	8.1 Interpretation of Results	91
8.1.1	8.1.1 The Text Dominance Phenomenon	91
8.1.2	8.1.2 Why Visual Features Do Not Help	92
8.1.3	8.1.3 The Role of Greek-Specific Pre-training	92
8.1.4	8.1.4 Implications for Multimodal Entity Resolution	93
8.2	8.2 Comparison with Production System	93
8.3	8.3 Limitations	95
8.3.1	8.3.1 Data Limitations	95
8.3.2	8.3.2 Methodological Limitations	95
8.3.3	8.3.3 Generalizability	96
8.4	8.4 Threats to Validity	96

8.4.1	8.4.1 Internal Validity	96
8.4.2	8.4.2 External Validity	97
9	Chapter 9: Conclusions and Future Work	97
9.1	9.1 Summary	97
9.1.1	9.1.1 Answers to Research Questions	97
9.2	9.2 Key Findings	98
9.3	9.3 Contributions	99
9.4	9.4 Future Work	99
9.4.1	9.4.1 Data Extensions	99
9.4.2	9.4.2 Methodological Extensions	100
9.4.3	9.4.3 Production Deployment	100
9.5	9.5 Closing Remarks	100
	References	101
	Appendix A: Greek Text Normalization	106
9.6	A.1 OCR Text Sanitization	106
9.7	A.2 Greek Normalization	107
9.8	A.3 Verification	108
	Appendix B: Detailed Results Tables	108
	Appendix C: Qualitative Examples	109
9.9	C.1 Hard Items Where Neural Models Succeed	109
9.10	C.2 Failure Cases	111
9.11	C.3 Image Feature Visualization	113
	Author's Statement	113

Abstract

Entity resolution — the task of determining whether two data representations refer to the same real-world entity — is a fundamental problem in data integration. In the retail domain, a critical instance of this problem arises when matching noisy OCR text extracted from purchase receipts to canonical product entries in a structured catalog. Current production systems rely predominantly on text-based fuzzy matching techniques such as BM25 and Elasticsearch, which struggle with abbreviated, misspelled, or truncated receipt text commonly produced by optical character recognition.

This thesis investigates whether a **multimodal approach** that combines textual and visual information can improve entity resolution accuracy for retail receipt line items. We propose a contrastive learning framework that encodes receipt items (OCR text and/or line-item image crops) and catalog products into a shared embedding space, enabling matching via nearest-neighbor retrieval. We systematically compare ten experimental configurations spanning five categories: (i) statistical text baselines (TF-IDF, BM25), (ii) a commercial API baseline (OpenAI text-embedding-3-small), (iii) neural text-only models (GreekBERT, mMiniLM), (iv) image-only models (DiT), and (v) multimodal models with three fusion strategies — early, late, and hybrid — plus an integrated vision-language model (LayoutLMv3).

Experiments are conducted on a real-world dataset of receipt line items from a major Greek supermarket chain, restricted to the food product category. The dataset includes OCR text, numerical attributes, ground-truth product mappings verified by human operators, and cropped line-item images. All data are anonymized and no personally identifiable information is used.

Results on text-only models demonstrate that contrastive fine-tuning of GreekBERT achieves an MRR of 0.9834 and Hit Rate@1 of 97.3%, outperforming the best statistical baseline (TF-IDF, MRR=0.8126) by +17.1 percentage points and the commercial OpenAI text-embedding-3-small API (MRR=0.7683) by +21.5 percentage points. The multilingual mMiniLM achieves MRR=0.9761, confirming that domain-specific fine-tuning provides larger gains than language-specific pre-training. A probe experiment combining text with receipt line-item image features (DiT, late fusion) showed no meaningful improvement (+0.09pp MRR), demonstrating that visual information does not carry additional discriminative power for this task due to the uniform visual appearance of receipt text. This finding has practical value: near-optimal

matching accuracy is achievable with text-only models, without the complexity of image processing pipelines.

The contributions of this work are: (1) a systematic empirical comparison of unimodal vs. multimodal entity resolution on real retail receipt data, (2) an evaluation of three fusion strategies under identical conditions, (3) application to the under-represented Greek language, and (4) practical insights for improving production receipt-matching systems.

Keywords: Entity Resolution, Multimodal Learning, Contrastive Learning, Receipt Understanding, OCR, Product Matching, Greek NLP

Abstract (Περίληψη στα Ελληνικά)

Η αντιστοίχιση οντοτήτων (entity resolution) — η διαδικασία αναγνώρισης ότι δύο αναπαραστάσεις δεδομένων αναφέρονται στην ίδια οντότητα του πραγματικού κόσμου — αποτελεί θεμελιώδες πρόβλημα στην ενοποίηση δεδομένων. Στον τομέα του λιανικού εμπορίου, μια κρίσιμη εκδοχή αυτού του προβλήματος προκύπτει κατά την αντιστοίχιση θορυβώδους κειμένου OCR, που εξάγεται από αποδείξεις αγορών, με κανονικές εγγραφές προϊόντων σε δομημένο κατάλογο.

Η παρούσα διπλωματική εργασία διερευνά κατά πόσον μια **πολυτροπική προσέγγιση** που συνδυάζει κειμενική και οπτική πληροφορία μπορεί να βελτιώσει την ακρίβεια αντιστοίχισης γραμμών αποδείξεων σε προϊόντα. Προτείνεται ένα πλαίσιο αντιπαραβολικής μάθησης (contrastive learning) που κωδικοποιεί γραμμές αποδείξεων και προϊόντα καταλόγου σε κοινό χώρο αναπαράστασης. Συγκρίνονται συστηματικά δέκα πειραματικές διαμορφώσεις: στατιστικές βασικές γραμμές (TF-IDF, BM25), εμπορικό API (OpenAI text-embedding-3-small), νευρωνικά κειμενικά μοντέλα (GreekBERT, mMiniLM), μοντέλο εικόνας (DiT), και πολυτροπικά μοντέλα με τρεις στρατηγικές σύντηξης, καθώς και το ενοποιημένο μοντέλο LayoutLMv3.

Τα αποτελέσματα δείχνουν ότι η αντιπαραβολική εκπαίδευση του GreekBERT επιτυγχάνει MRR 0,9834 και Hit Rate@1 97,3%, ξεπερνώντας τη βέλτιστη στατιστική βασική γραμμή (TF-IDF, MRR=0,8126) κατά +17,1 ποσοστιαίες μονάδες και το εμπορικό API της OpenAI (MRR=0,7683) κατά +21,5 μονάδες. Το πολύγλωσσο mMiniLM επιτυγ-

χάνει $MRR=0,9761$, επιβεβαιώνοντας ότι η εξειδικευμένη εκπαίδευση υπερτερεί της γλωσσολογικής προ-εκπαίδευσης. Πείραμα-ανίχνευση (probe) που συνδυάζει κείμενο με οπτικά χαρακτηριστικά εικόνας (DiT, late fusion) δεν έδειξε ουσιαστική βελτίωση ($+0,09pp$ MRR), αποδεικνύοντας ότι η οπτική πληροφορία δεν συνεισφέρει επιπλέον διακριτική ικανότητα λόγω της ομοιόμορφης οπτικής εμφάνισης του κειμένου αποδείξεων.

Λέξεις-κλειδιά: Αντιστοίχιση Οντοτήτων, Πολυτροπική Μάθηση, Αντιπαραβολική Μάθηση, Κατανόηση Αποδείξεων, OCR, Αντιστοίχιση Προϊόντων, Ελληνική NLP

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dimitrios Karapiperis, for his guidance, constructive feedback, and support throughout this thesis. His insights helped shape the direction of this work and pushed me to think more critically about the research.

I would also like to thank the company that provided access to the receipt and product catalog data used in this study. Without their willingness to support academic research with real-world data, this work would not have been possible.

List of Tables

- Table 2.1: Comparison of traditional vs. neural entity resolution approaches
- Table 3.1: Summary of related work on multimodal entity resolution
- Table 4.1: Dataset statistics
- Table 4.2a: Easy matching examples (high lexical overlap)
- Table 4.2b: Hard matching examples (low or zero lexical overlap)
- Table 4.2c: Product catalog disambiguation challenge — near-identical entries
- Table 4.2d: Sample receipt walkthrough — from OCR to product matching
- Table 4.2e: Product category distribution in the food subtree
- Table 4.3: OCR text length statistics
- Table 4.4: Train / Validation / Test split sizes

- Table 4.5: Text normalization examples
- Table 5.1: Experiment matrix — all ten configurations
- Table 6.1: Hyperparameter settings for all models
- Table 6.2: Pre-trained model specifications
- Table 7.1: Main results — all models compared on test set
- Table 7.2a: Epoch-by-epoch training dynamics for GreekBERT (T1)
- Table 7.2b: Epoch-by-epoch training dynamics for mMiniLM (T2)
- Table 7.3: Detailed retrieval metrics at K=1, 5, 10
- Table 7.4: Top 10 product categories in the test set
- Table 7.5: Ablation study — embedding dimension
- Table 7.6: Ablation study — temperature parameter
- Table 7.7: Test set distribution by OCR abbreviation level
- Table 7.7a: Representative examples by OCR abbreviation tier
- Table 7.8: Image probe results
- Table B.1: Full per-experiment metrics with confidence intervals

List of Figures

- Figure 1.1: Receipt line item matching problem illustration
- Figure 2.1: Entity resolution pipeline overview
- Figure 2.2: Transformer encoder architecture
- Figure 2.3: Contrastive learning in shared embedding space
- Figure 4.1: Distribution of receipt items per product
- Figure 4.2: OCR text length histogram
- Figure 4.3: Product category tree (food subtree excerpt)
- Figure 4.4: Receipt photograph and extracted line-item crops
- Figure 4.4b: Google Vision OCR bounding box detection on receipt image
- Figure 4.5: OCR text normalization pipeline — two-stage processing from raw receipt text to normalized input
- Figure 5.1: Overall system architecture
- Figure 5.2: Early fusion architecture
- Figure 5.3: Late fusion architecture
- Figure 5.4: Hybrid fusion architecture with cross-attention

- Figure 5.5: LayoutLMv3 integrated architecture
 - Figure 7.1a: Main results — model comparison across evaluation metrics
 - Figure 7.1b: MRR comparison with 95% confidence intervals
 - Figure 7.1c: Hit Rate at different K values
 - Figure 7.1d: Precision@K and Recall@K curves
 - Figure 7.2: Training loss and validation MRR curves
 - Figure 7.4–7.5: Ablation studies — embedding dimension and temperature
 - Figure 7.8: Image probe results — text-only vs. text+image
 - Figure 8.1: Progressive MRR improvement across methods (waterfall)
-

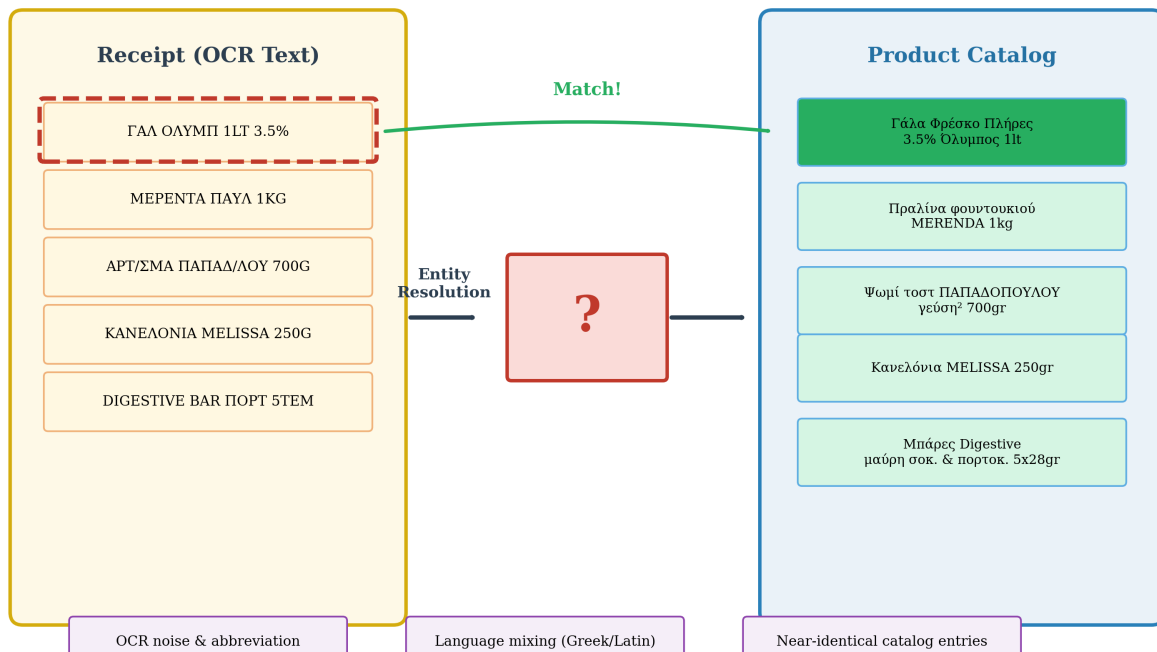
1 Chapter 1: Introduction

1.1 1.1 Background and Motivation

The proliferation of digital receipt management applications has created an unprecedented need for automated systems that can accurately interpret and structure the information contained in purchase receipts. When a consumer photographs a receipt from a retail store, Optical Character Recognition (OCR) technology extracts the text, producing a sequence of line items. Each line item typically contains an abbreviated product description, quantity, and price. The critical challenge lies in matching these noisy, abbreviated text strings to the correct entries in a canonical product catalog — a task known as **entity resolution** (Christophides et al., 2021).

Entity resolution (ER), also referred to as record linkage, entity matching, or deduplication, is a long-standing problem in database research (Elmagarmid et al., 2007). The core question is: given two records from different data sources, do they refer to the same real-world entity? In the retail receipt domain, this translates to: does the OCR text ΓΑΛ ΟΛΥΜΠ 1LT 3.5% refer to the product Γάλα Φρέσκο Πλήρες 3.5% Όλυμπος 1lt in the catalog?

Figure 1.1: The receipt-to-product matching problem. A receipt line item containing abbreviated OCR text (e.g., “ΓΑΛ ΟΛΥΜΠ 1LT 3.5%”) must be matched to the correct entry in a product catalog containing thousands of products (e.g., “Γάλα Φρέσκο Πλήρες 3.5% Όλυμπος 1lt”). The matching is complicated by OCR noise, abbreviations, language mixing,



Σχήμα 1: Figure 1.1: Receipt line item matching problem

and near-identical catalog entries.

This matching task is particularly challenging for several reasons:

1. **OCR noise:** Character recognition errors introduce misspellings, merged words, and character substitutions (e.g., 0 for O, 1 for I).
2. **Abbreviation:** Receipt printers impose strict character limits, forcing product names to be severely truncated (e.g., ΓΑΛΑ → ΓΑΛ).
3. **Ambiguity:** Many products have nearly identical names, differing only in attributes like fat percentage, package size, or flavor variant.
4. **Greek language specifics:** The Greek alphabet, diacritics (accents), and inflectional morphology introduce additional complexity for text processing systems primarily designed for Latin-script languages.

Current production systems, including the one studied in this work, rely on text-based approaches — primarily Elasticsearch with fuzzy matching and BM25 scoring (Robertson & Zaragoza, 2009). While these methods achieve reasonable accuracy, they are inherently limited to the textual modality and cannot leverage visual cues present in the receipt image itself.

1.2 1.2 Research Problem

This thesis addresses the following research problem:

Can a multimodal entity resolution approach that jointly leverages OCR text and receipt line-item images achieve higher product matching accuracy than text-only methods?

The receipt image contains information beyond what OCR extracts: character spacing, font characteristics, line structure, alignment patterns, and visual context from adjacent items. We hypothesize that a model capable of processing both modalities can capture complementary signals, leading to improved matching performance — particularly for ambiguous or degraded OCR cases.

1.3 1.3 Research Questions

The following research questions guide this investigation:

RQ1: How do neural text encoding methods (GreekBERT, mMiniLM) compare to traditional statistical methods (TF-IDF, BM25) for receipt-to-product matching in Greek?

RQ2: Can visual information from receipt line-item images, processed independently of text, provide sufficient signal for product identification?

RQ3: Does combining textual and visual modalities improve matching accuracy over the best unimodal approach, and which fusion strategy (early, late, or hybrid) is most effective?

RQ4: How does an integrated vision-language model (LayoutLMv3) that jointly processes text, image, and layout information compare to separate-encoder fusion approaches?

1.4 1.4 Approach Overview

We frame receipt-to-product matching as an **information retrieval** problem: given a query (receipt line item), rank all products in the catalog by likelihood of being the correct match. This formulation enables the use of standard retrieval metrics and fair comparison across methods.

All neural models are trained using **contrastive learning** (Oord et al., 2018), specifically InfoNCE loss, which learns to place matching receipt-product pairs close together in a shared

embedding space while pushing non-matching pairs apart. At inference time, matching is performed via efficient nearest-neighbor search using FAISS (Johnson et al., 2019).

Ten experimental configurations are evaluated, organized in five categories:

- **Statistical baselines:** TF-IDF, BM25
- **Neural text-only:** GreekBERT (Koutsikakis et al., 2020), mMiniLM (Wang et al., 2020)
- **Image-only:** DiT (Li et al., 2022a)
- **Multimodal:** Early fusion, Late fusion, Hybrid fusion (cross-attention), LayoutLMv3 (Huang et al., 2022)

1.5 1.5 Contributions

This thesis makes the following contributions:

1. **Systematic multimodal comparison:** A controlled experimental comparison of text-only, image-only, and multimodal approaches for receipt entity resolution, using identical data splits and evaluation protocol.
2. **Principled multimodal evaluation methodology:** A rigorous approach to deciding when multimodal fusion is warranted, using a targeted image probe experiment to establish whether visual features carry discriminative signal before committing to full multimodal training.
3. **Greek-language evaluation:** Application and evaluation of multilingual and Greek-specific language models on receipt text, contributing to the under-explored area of Greek NLP for domain-specific tasks.
4. **Practical insights:** Analysis of when and why multimodal approaches succeed or fail, providing actionable guidance for production receipt-matching systems.

1.6 1.6 Thesis Structure

The remainder of this thesis is organized as follows:

- **Chapter 2** establishes the theoretical background, covering entity resolution, transformer architectures, contrastive learning, and multimodal fusion.
- **Chapter 3** reviews related work in entity resolution, multimodal matching, document

understanding, and receipt processing.

- **Chapter 4** describes the dataset: its source, composition, anonymization, and partitioning.
 - **Chapter 5** presents the proposed methodology, including all model architectures and the training procedure.
 - **Chapter 6** details the experimental setup: implementation, hyperparameters, hardware, and evaluation protocol.
 - **Chapter 7** reports the experimental results, including the main comparison, ablation studies, and error analysis.
 - **Chapter 8** discusses the findings, their implications, limitations, and threats to validity.
 - **Chapter 9** concludes the thesis and outlines directions for future work.
-

2 Chapter 2: Theoretical Background

2.1 Entity Resolution

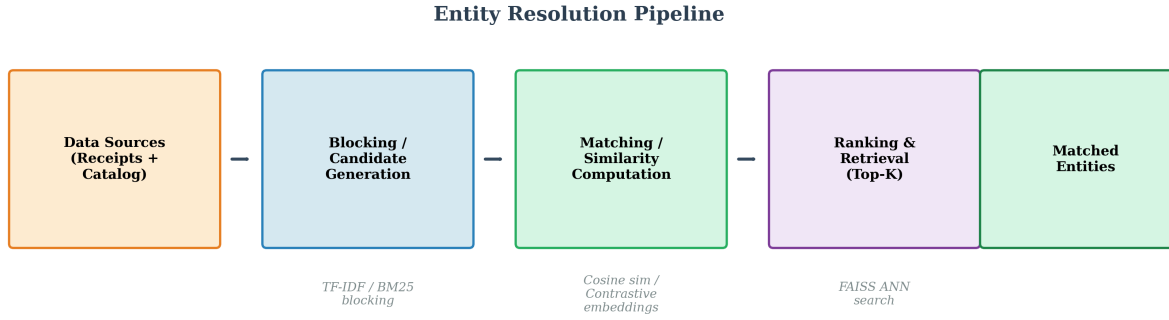
2.1.1 Problem Definition

Entity Resolution (ER) is the process of identifying records in one or more datasets that refer to the same real-world entity (Christophides et al., 2021). Formally, given two record sets R_A and R_B , the goal is to find all pairs $(r_a, r_b) \in R_A \times R_B$ such that r_a and r_b represent the same entity. In our setting, R_A is the set of receipt line items and R_B is the product catalog.

ER is closely related to several other tasks in the literature:

- **Record Linkage** (Fellegi & Sunter, 1969): The classical statistical framework for linking records across databases.
- **Deduplication**: Identifying duplicate records within a single dataset.
- **Entity Matching**: A binary classification formulation — given a pair of records, predict whether they match.

Figure 2.1: Entity resolution pipeline overview. The standard ER pipeline consists of four stages: data source ingestion, blocking/candidate generation to reduce the comparison



Σχήμα 2: Figure 2.1: Entity resolution pipeline overview

space, similarity computation between candidate pairs, and ranking with top-K retrieval. In our framework, blocking is implicit (all products are candidates), similarity is computed via contrastive embeddings, and retrieval uses FAISS approximate nearest-neighbor search.

2.1.2 Traditional Approaches

Traditional ER approaches operate in three phases (Papadakis et al., 2021):

1. **Blocking:** Reduce the quadratic comparison space by grouping records into candidate blocks using cheap heuristics (e.g., shared tokens, phonetic codes).
2. **Matching:** Compute similarity between candidate pairs using string similarity metrics (Levenshtein distance, Jaccard similarity, TF-IDF cosine) or learned classifiers.
3. **Clustering:** Resolve transitive matches and conflicts to produce final entity clusters.

In the product matching context, text-based similarity measures have been the dominant approach:

- **TF-IDF** (Salton & Buckley, 1988): Represents text as weighted term-frequency vectors, with inverse document frequency downweighting common terms. Similarity is computed via cosine distance.
- **BM25** (Robertson & Zaragoza, 2009): A probabilistic relevance model that extends TF-IDF with document length normalization and term frequency saturation. It remains a strong baseline in information retrieval.

Table 2.1: Comparison of traditional vs. neural entity resolution approaches

Aspect	Traditional (TF-IDF/BM25)	Neural (Transformer-based)
Representation	Sparse, term-frequency	Dense, contextual embeddings
Similarity	Lexical overlap	Semantic similarity
Abbreviations	Poor (no shared terms)	Better (learned subword patterns)
Training data	None required	Requires labeled pairs
Scalability	Very fast	Requires GPU, but scales with FAISS
Interpretability	High (term weights)	Low (black-box embeddings)

2.1.3 Neural Entity Resolution

Recent advances in deep learning have transformed entity resolution. Neural approaches learn dense vector representations (embeddings) of records, enabling semantic matching beyond lexical overlap (Mudgal et al., 2018; Li et al., 2020).

Key advantages of neural ER include:

- **Handling abbreviations:** Subword tokenization (e.g., WordPiece, BPE) can match partial words that share subword units.
- **Contextual understanding:** Transformer-based encoders capture context-dependent meaning, distinguishing “1LT” (1 liter) from “1LT” in other contexts.
- **Transfer learning:** Pre-trained language models bring general linguistic knowledge that can be fine-tuned for domain-specific matching.

The dominant paradigm is to fine-tune pre-trained transformer models on labeled entity pairs using either: - **Binary classification:** Predict match/no-match for each pair (Brunner & Stockinger, 2020). - **Contrastive learning:** Learn an embedding space where matching entities are close and non-matching entities are far (Li et al., 2020).

2.2 Transformer Architecture

2.2.1 Self-Attention Mechanism

The Transformer architecture (Vaswani et al., 2017) is the foundation of modern NLP and multimodal models. Its core innovation is the **self-attention mechanism**, which allows each token in a sequence to attend to all other tokens, capturing long-range dependencies without

recurrence.

Given an input sequence of token embeddings $X \in \mathbb{R}^{n \times d}$, self-attention computes:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

where $Q = XW_Q$, $K = XW_K$, $V = XW_V$ are linear projections of the input into query, key, and value spaces respectively, and d_k is the dimension of the key vectors.

Figure 2.2: Transformer encoder architecture. Each encoder layer consists of multi-head self-attention followed by a feed-forward network, with residual connections and layer normalization after each sub-layer. The encoder is applied N times ($N=12$ for BERT-base models). The output contextual embeddings are aggregated via mean pooling to produce a fixed-dimensional representation for each input sequence.

Multi-head attention extends this by computing attention in parallel across h heads with different projections, then concatenating and projecting the results:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W_O$$

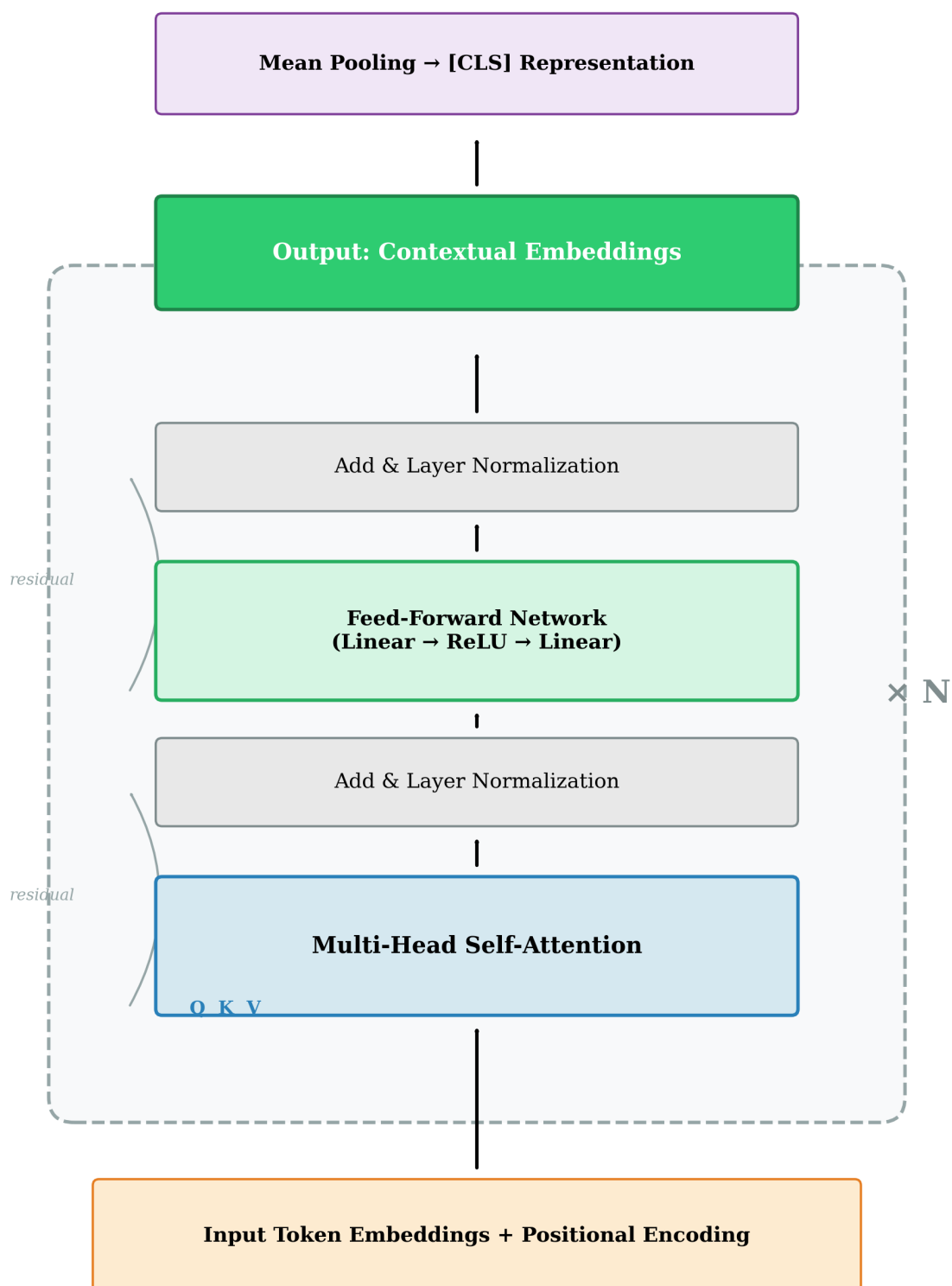
2.2.2 BERT and Pre-trained Language Models

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) introduced a pre-training paradigm where a transformer encoder is trained on large text corpora using two objectives:

1. **Masked Language Modeling (MLM):** Randomly mask 15% of input tokens and predict them from context.
2. **Next Sentence Prediction (NSP):** Predict whether two sentences are consecutive in the original text.

The resulting model captures deep bidirectional contextual representations that can be fine-tuned for downstream tasks. For entity resolution, the [CLS] token representation or mean pooling of token representations serves as the record embedding.

GreekBERT (Koutsikakis et al., 2020) adapts this approach to Greek by pre-training on a large Greek text corpus, using a Greek-specific vocabulary. This is particularly relevant for our task,



Σχήμα 3: Figure 2.2: Transformer encoder architecture

as Greek receipt text contains domain-specific vocabulary and abbreviation patterns that benefit from language-specific pre-training.

2.2.3 2.2.3 Efficient Transformers and Knowledge Distillation

Full-scale BERT models (110M+ parameters) can be prohibitively expensive for deployment. Knowledge distillation techniques compress large models into smaller ones that retain most of the performance:

- **MiniLM** (Wang et al., 2020): Distills the self-attention distributions of the teacher model, producing compact models that preserve attention behavior. The multilingual variant **mMiniLM** extends this to 100+ languages, including Greek.
- **Sentence-BERT** (Reimers & Gurevych, 2019): Modifies BERT with siamese and triplet network structures to produce semantically meaningful sentence embeddings efficient for similarity search.

2.3 2.3 Vision Transformers and Document Understanding

2.3.1 2.3.1 Vision Transformer (ViT)

The Vision Transformer (Dosovitskiy et al., 2021) applies the transformer architecture to images by splitting an image into fixed-size patches, linearly projecting each patch into a token embedding, and processing the sequence with a standard transformer encoder. This approach has achieved state-of-the-art results on image classification, rivaling CNNs.

2.3.2 2.3.2 Document Image Transformer (DiT)

DiT (Document Image Transformer) (Li et al., 2022a) is a self-supervised pre-trained model specifically designed for document image understanding. It applies the ViT architecture to document images, pre-trained using a masked image modeling objective on the IIT-CDIP dataset of 42 million document images.

DiT is particularly relevant for our work because: - It is pre-trained on **document images**, not natural photographs, so its learned representations capture text layout, font patterns, and document structure. - It can process receipt line-item crops, extracting visual features that complement OCR text.

2.3.3 2.3.3 LayoutLMv3

LayoutLMv3 (Huang et al., 2022) represents the state of the art in multimodal document understanding. Unlike separate text and image encoders, LayoutLMv3 jointly pre-trains on text, image, and layout (spatial position) information using:

1. **Masked Language Modeling:** Predicting masked text tokens from context and layout.
2. **Masked Image Modeling:** Predicting masked image patches.
3. **Word-Patch Alignment:** Predicting whether a text token and image patch correspond to the same location.

This unified approach captures the inherent alignment between text content and its visual appearance in documents, making it a natural fit for receipt understanding where spatial layout carries information.

2.4 2.4 Contrastive Learning

2.4.1 2.4.1 Contrastive Learning Framework

Contrastive learning is a self-supervised or supervised learning paradigm that learns representations by contrasting positive pairs (similar items) against negative pairs (dissimilar items) (Le-Khac et al., 2020). The objective is to learn an embedding space where semantically similar items are close and dissimilar items are far apart.

2.4.2 2.4.2 InfoNCE Loss

The InfoNCE loss (Oord et al., 2018), also known as NT-Xent (Normalized Temperature-scaled Cross-Entropy), is the most widely used contrastive loss function. For a positive pair (x_i, x_i^+) within a batch of N pairs:

$$\mathcal{L}_i = -\log \frac{\exp(\text{sim}(z_i, z_i^+) / \tau)}{\sum_{j=1}^N \exp(\text{sim}(z_i, z_j^+) / \tau)}$$

where $z_i = f(x_i)$ and $z_j^+ = g(x_j^+)$ are the normalized embeddings produced by the query and key encoders respectively, $\text{sim}(\cdot, \cdot)$ is cosine similarity, and τ is a temperature parameter controlling the sharpness of the distribution.

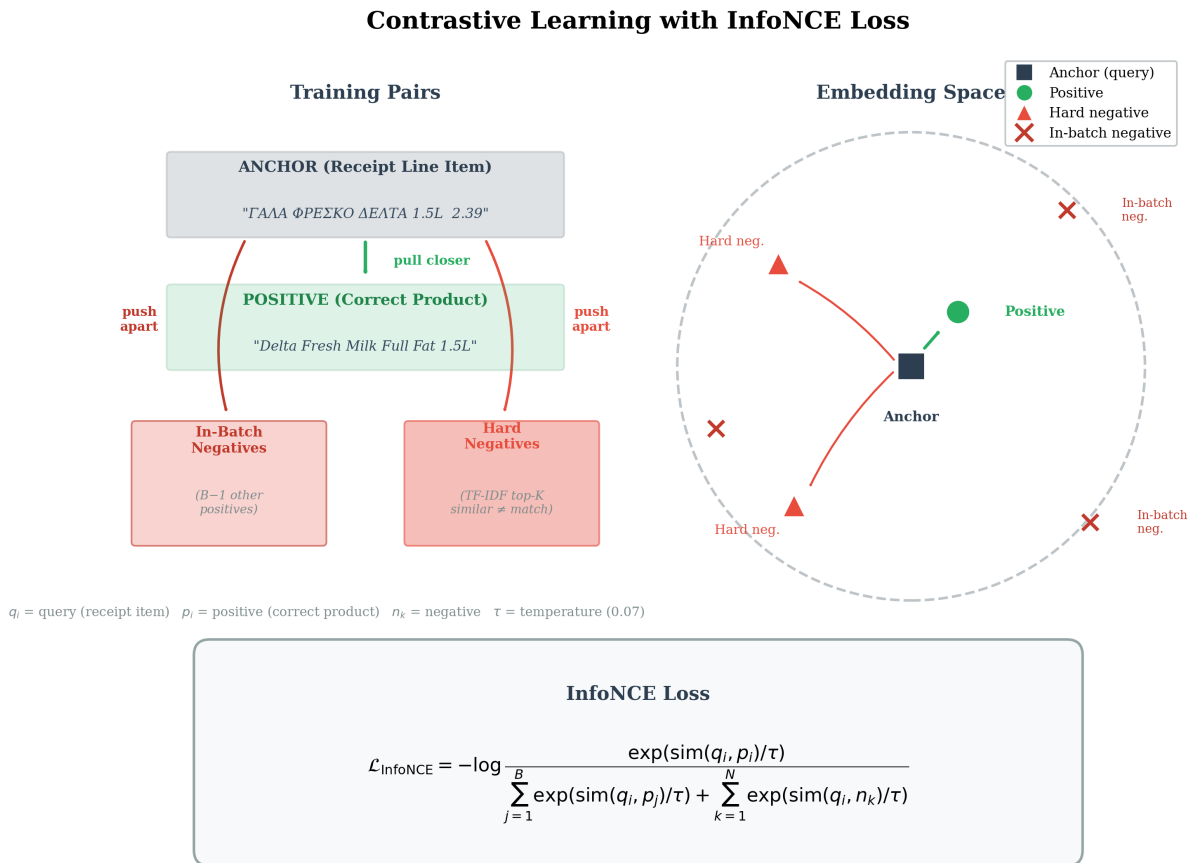
The temperature τ plays a critical role: - **Low τ** (e.g., 0.01): Sharp distribution, focuses on

hard negatives but risks training instability. - **High τ** (e.g., 1.0): Uniform distribution, treats all negatives equally, leading to less discriminative embeddings. - **Typical values:** $\tau \in [0.05, 0.1]$ for most contrastive learning applications (Chen et al., 2020).

2.4.3 Hard Negative Mining

The quality of negative pairs is critical for contrastive learning. **Easy negatives** (clearly different products) provide little learning signal, while **hard negatives** (similar but different products) force the model to learn fine-grained distinctions (Robinson et al., 2021).

In our product matching context, hard negatives are products that share significant textual overlap with the correct match — for example, different variants of the same brand’s milk. We employ a strategy of mining hard negatives from within the same product subcategory.



Σχήμα 4: Figure 2.3: Contrastive learning workflow

Figure 2.3: Contrastive learning with InfoNCE loss. The left panel illustrates the training pair structure: each receipt line item (anchor) is paired with its correct catalog product (positive) and contrasted against in-batch negatives (other products in the mini-batch) and hard

negatives (similar but incorrect products mined via TF-IDF similarity). The right panel shows the resulting embedding space: the loss function pulls the anchor closer to the positive while pushing negatives away. Hard negatives — products that are textually similar to the correct match but refer to different entities — are particularly important as they force the model to learn fine-grained distinctions, such as differentiating between milk variants of the same brand.

2.5 2.5 Multimodal Fusion Strategies

When combining information from multiple modalities (text and image in our case), the choice of **fusion strategy** determines how and when the modalities interact (Baltrusaitis et al., 2019):

2.5.1 2.5.1 Early Fusion

In early fusion, the feature representations from each modality are concatenated before any joint processing:

$$z_{fused} = \text{MLP}([\text{enc}_T(x_T); \text{enc}_I(x_I)])$$

Advantages: Allows cross-modal interaction from the start; captures fine-grained correlations.

Disadvantages: Requires modalities to have compatible representations; more parameters.

2.5.2 2.5.2 Late Fusion

In late fusion, each modality produces an independent similarity score, and the final score is a weighted combination:

$$s_{fused} = \alpha \cdot s_T + (1 - \alpha) \cdot s_I$$

Advantages: Modular; each modality can be pre-trained independently; robust when one modality is degraded. **Disadvantages:** Cannot capture cross-modal correlations; assumes modality independence.

2.5.3 2.5.3 Hybrid Fusion (Cross-Attention)

Hybrid fusion uses a cross-attention mechanism where one modality attends to the other, allowing information exchange while maintaining modality-specific representations:

$$z'_T = \text{CrossAttention}(Q = z_T, K = z_I, V = z_I)$$

$$z'_I = \text{CrossAttention}(Q = z_I, K = z_T, V = z_T)$$

$$z_{\text{used}} = \text{MLP}([z'_T; z'_I])$$

Advantages: Captures cross-modal interactions while preserving modality-specific information. **Disadvantages:** Most complex; more parameters and computation; risk of overfitting with limited data.

2.6 2.6 Approximate Nearest Neighbor Search

Efficient retrieval over large embedding spaces requires approximate nearest neighbor (ANN) search. **FAISS** (Facebook AI Similarity Search) (Johnson et al., 2019) is a library of algorithms for dense vector search that supports:

- **Flat index** (exact search): Brute-force comparison, exact results, $O(n \cdot d)$ per query.
- **IVF (Inverted File)**: Clusters vectors and only searches nearby clusters, reducing search time.
- **HNSW (Hierarchical Navigable Small World)**: Graph-based search with logarithmic complexity.

For our product catalog sizes (thousands to tens of thousands of products), flat exact search is sufficiently fast. FAISS is used primarily for its efficient batch similarity computation during evaluation.

3 Chapter 3: Related Work

3.1 3.1 Entity Resolution with Deep Learning

The application of deep learning to entity resolution has gained significant traction in recent years. **DeepER** (Ebraheem et al., 2018) was among the first works to apply distributed word representations (word2vec, GloVe) for entity matching, showing that even simple aggregations

of pre-trained embeddings outperform hand-crafted string similarity features.

DeepMatcher (Mudgal et al., 2018) introduced a comprehensive framework for deep entity matching, proposing modular architectures with attribute summarization, attribute comparison, and classification components. Their work demonstrated that attention-based models significantly outperform simple aggregation, especially for “dirty” data with textual noise — a finding directly relevant to OCR receipt text.

Ditto (Li et al., 2020) advanced the state of the art by fine-tuning pre-trained language models (BERT, RoBERTa) for entity matching. Key innovations include domain knowledge injection through data augmentation (span deletion, attribute manipulation) and training optimization through summarization of long inputs. Ditto achieved state-of-the-art results on multiple ER benchmarks, establishing pre-trained transformers as the dominant approach.

More recently, **PromptEM** (Wang, Zeng, et al., 2023) explored prompt-based approaches for entity matching, reformulating ER as a conditional text generation task using large language models. While promising, such approaches require significantly more computational resources than the contrastive learning framework we adopt.

3.2 Multimodal Entity Resolution

Multimodal entity resolution extends traditional text-based ER by incorporating additional modalities, most commonly images. This direction has been explored primarily in the **e-commerce product matching** domain.

Wilke & Rahm (2021) proposed a multimodal approach for product entity resolution that extends attribute-based matching with image data. Their work demonstrated that combining product images with textual descriptions improves match quality, particularly for visually distinctive products where textual attributes alone are insufficient for reliable match decisions. This is one of the earliest works to systematically evaluate the benefit of adding visual features to entity resolution in an e-commerce context.

MMEA (Multimodal Entity Alignment) (Guo et al., 2021) explored multimodal entity alignment in knowledge graphs, combining structural, textual, and visual features. While not directly applicable to receipt matching, their cross-modal attention mechanism for fusing textual and visual features informed our hybrid fusion design.

ProductMatch (Tracz et al., 2020) specifically addressed product entity matching using product images and titles from online marketplaces. Their dual-encoder architecture with late fusion demonstrated that images are particularly helpful when product titles are ambiguous or incomplete — a scenario analogous to abbreviated receipt text.

Our work differs from these prior approaches in several important ways:

1. **Domain:** We work with OCR-extracted receipt text rather than product listings — the text is significantly noisier and more abbreviated.
2. **Visual input:** Rather than product photographs, we use **receipt line-item image crops**, which contain typographic rather than photographic information.
3. **Language:** We specifically address the challenges of Greek-language receipt text.
4. **Retrieval formulation:** We adopt a contrastive retrieval approach rather than binary classification, enabling efficient real-time matching.

3.3 Receipt and Document Understanding

Receipt understanding has been studied as a specific instance of document understanding. The **CORD** (Consolidated Receipt Dataset) benchmark (Park et al., 2019) introduced a standardized evaluation for receipt information extraction, covering key fields like store name, date, item names, and totals.

SROIE (Scanned Receipts OCR and Information Extraction) (Huang et al., 2019) is another widely used benchmark that focuses on OCR and key information extraction from receipts. Both benchmarks address information extraction rather than entity resolution, but they demonstrate the importance of combining visual and textual features for receipt processing.

In the broader document understanding literature, **LayoutLM** (Xu et al., 2020) pioneered the joint pre-training of text and layout for document understanding tasks. Its successors **LayoutLMv2** (Xu et al., 2021) and **LayoutLMv3** (Huang et al., 2022) progressively added visual features and improved pre-training objectives, achieving state-of-the-art results on document understanding benchmarks including receipt analysis.

DocFormer (Appalaraju et al., 2021) proposed a multi-modal transformer for document understanding that separately encodes text, visual, and spatial features before fusing them via shared self-attention layers. **SelfDoc** (Li et al., 2021) introduced a self-supervised pre-training

approach for document understanding that learns cross-modal feature interactions.

3.4 3.4 Contrastive Learning for Retrieval

Contrastive learning has become the dominant paradigm for learning retrieval-oriented embeddings. **DPR** (Dense Passage Retrieval) (Karpukhin et al., 2020) demonstrated that contrastive fine-tuning of BERT for passage retrieval significantly outperforms BM25, establishing the viability of dense retrieval for information retrieval tasks.

CLIP (Contrastive Language-Image Pre-training) (Radford et al., 2021) scaled contrastive learning to 400 million image-text pairs, learning a shared embedding space for text and images. CLIP’s success demonstrated that contrastive learning is effective for aligning representations across modalities — a key motivation for our multimodal approach.

SimCLR (Chen et al., 2020) provided a systematic study of contrastive learning design choices, showing that: (i) composition of data augmentations is critical, (ii) a learnable nonlinear projection head improves representation quality, and (iii) larger batch sizes improve performance by providing more negative pairs. We incorporate these insights into our training procedure.

E5 (Wang et al., 2022) demonstrated that contrastive pre-training combined with text-pair supervision produces high-quality text embeddings that transfer well to retrieval tasks, achieving strong performance across multiple benchmarks including multilingual settings.

3.5 3.5 Greek Natural Language Processing

Greek is considered a **mid-resource language** in the NLP landscape — it has more resources than truly low-resource languages but significantly fewer than English, Chinese, or other high-resource languages.

GreekBERT (Koutsikakis et al., 2020) is a BERT-base model pre-trained on a 29GB Greek corpus comprising web text, Wikipedia, and European Parliament proceedings. It uses a Greek-specific WordPiece vocabulary of 35,000 tokens, enabling better tokenization of Greek text compared to multilingual models like mBERT.

Greek-BERT evaluation studies (Otsios et al., 2020) have shown that language-specific models consistently outperform multilingual alternatives for Greek text classification, named

entity recognition, and sentiment analysis tasks. This suggests that GreekBERT is preferable to mMiniLM for tasks where Greek text understanding is central, though the performance gap varies by task.

For our specific domain of receipt text, no prior work has evaluated these models. Receipt text is substantially different from the web and news text on which these models were pre-trained: it is shorter, more abbreviated, uses all-caps, and contains domain-specific product nomenclature.

3.6 3.6 Product Matching in E-commerce

Product matching and deduplication in e-commerce has been extensively studied due to its commercial importance.

Köpcke et al. (2010) provided an early evaluation framework for product matching, comparing multiple approaches on benchmark datasets from online shops. Their analysis highlighted the importance of domain-specific blocking strategies for scalability.

Shah et al. (2018) introduced a multi-task learning approach for product matching that jointly learns matching and attribute extraction, demonstrating that auxiliary tasks improve matching accuracy. Their work on Amazon product data showed that product category information is a valuable signal for disambiguation.

MAVE (Yang et al., 2022) proposed a large-scale dataset and attribute extraction system for e-commerce product matching, demonstrating that structured attribute extraction (brand, size, type) can improve matching accuracy over holistic text comparison. This insight is relevant to our work, as receipt text often encodes attributes (weight, fat percentage) that could be explicitly modeled.

3.7 3.7 Summary and Research Gap

Table 3.1 summarizes the most relevant prior work and positions our contribution:

Table 3.1: Summary of related work on multimodal entity resolution

Work	Modalities	Domain	Language	Matching Approach
DeepMatch (Mudgal et al., 2018)	Text	General ER benchmarks	English	Attention-based classification
Ditto (Li et al., 2020)	Text	General ER benchmarks	English	Fine-tuned BERT classification
Wilke & Rahm (2021)	Text + Image	E-commerce	Multi	Multimodal attribute matching
ProductMatch (Tracz et al., 2020)	Text + Image	E-commerce	English	Late fusion dual-encoder
LayoutLMv2 (Huang et al., 2022)	Text + Image + Layout	Document understanding	English	Unified transformer
This work	Text + Image	Retail receipts	Greek	Contrastive retrieval + 3 fusion strategies

3.8 Recent Developments: LLMs for Entity Resolution

Since the experimental work in this thesis was conducted, the rapid evolution of large language models (LLMs) has opened new directions for entity resolution.

Peeters, Steiner, & Bizer (2025) conducted a comprehensive evaluation of generative LLMs for entity matching, showing that GPT-4 outperforms the best pre-trained language model (PLM) baselines by 40–68% F1 on standard ER benchmarks. Their results suggest that LLMs can serve as zero-shot or few-shot matchers without task-specific training, potentially eliminating the need for labeled data. However, the computational cost and latency of LLM-

based matching (hundreds of milliseconds per pair) remain prohibitive for real-time production systems processing thousands of queries per second.

DocLLM (Wang et al., 2024) extends the document understanding paradigm beyond LayoutLMv3 by incorporating layout-aware attention into a generative LLM framework. Unlike LayoutLMv3, DocLLM avoids expensive image encoders and relies solely on bounding box information to capture spatial structure. This approach achieved strong results on document understanding benchmarks and could potentially be applied to receipt understanding, though its generative nature makes it less suitable for the retrieval formulation adopted in this thesis.

Cross-lingual product matching has also seen recent advances. Peeters et al. (2024) explored cross-lingual learning strategies for product matching, demonstrating that transfer learning from high-resource languages can improve matching quality in low-resource settings. This is relevant to our work, as Greek receipt text frequently mixes Greek and Latin scripts — a challenge that cross-lingual approaches could address more explicitly.

These developments suggest that future work on receipt entity resolution could benefit from LLM-based approaches for handling the most difficult matching cases (extreme abbreviation, zero lexical overlap), while maintaining the efficiency of contrastive retrieval for the majority of queries.

The research gap addressed by this thesis is the intersection of (i) multimodal entity resolution with (ii) receipt line-item matching in (iii) a non-English language setting. No prior work has explored the combination of OCR text and receipt image crops for product matching using contrastive learning, nor has the effectiveness of multimodal approaches been evaluated for Greek retail data.

4 Chapter 4: Data Description

4.1 4.1 Data Source

The data used in this study originate from a Greek retail technology platform that processes consumer purchase receipts. The platform operates a receipt scanning system where consumers

photograph their receipts using a mobile application. The system performs OCR to extract text, then matches each line item to canonical products in a curated catalog.

The platform maintains two database systems:

1. **Receipt database:** Stores raw receipt data, OCR results, and line-item groupings.
2. **Product catalog database:** Stores the canonical product catalog with product names, brands, categories, and attributes.

All data used in this study were exported via read-only SQL queries from a staging environment. No production data was accessed, and no data was modified during export.

4.2 4.2 Scope and Selection Criteria

To ensure a focused and manageable experimental scope, the following selection criteria were applied:

- **Retailer:** A single major Greek supermarket chain (anonymized as “Retailer A”). Restricting to one retailer ensures consistent receipt format, OCR quality, and product naming conventions.
- **Product category:** The **food** (Τρόφιμα) category subtree only. Food is the largest product category, providing sufficient data volume while maintaining thematic coherence. It also presents the most challenging matching scenarios, with many similar products (e.g., multiple milk brands, yogurt variants).
- **Time period:** One calendar month (February 2026) of receipts. This provides a representative sample of typical purchasing patterns while maintaining temporal homogeneity.
- **Quality filters:** Only receipts that were successfully analyzed (OCR completed) and line items with non-empty OCR text that have been assigned to a product group with a confirmed product match.

4.3 4.3 Dataset Components

4.3.1 4.3.1 Receipt Line Items

Each receipt line item contains:

Field	Description	Example
ocr_text	Raw text extracted by OCR	ΓΑΛ ΟΛΥΜΠ 1LT 3.5%
is_weighted	Whether the product is sold by weight	false
quantity	Number of units purchased	2
unit_price	Price per unit	1.89
final_price	Total price for this line	3.78

The ocr_text field is the primary textual input. It is typically 10-40 characters long, uses Greek uppercase letters, and frequently contains abbreviations, truncations, and OCR errors.

4.3.2 4.3.2 Product Catalog

The canonical product catalog contains:

Field	Description	Example
name	Full product name	Γάλα Φρέσκο Πλήρες 3.5% Όλυμπος 1lt
brand_name	Brand name	Όλυμπος
product_type_name	Category name	Γάλατα Φρέσκα
is_weighted	Whether sold by weight	false
uom	Unit of measurement	lt
uom_value	Measurement value	1

Product names are properly formatted Greek text with mixed case, full brand names, and structured attribute information.

4.3.3 4.3.3 Ground Truth

The ground truth mapping connects receipt line items to canonical products through an intermediate grouping mechanism:

1. The system assigns each line item to a **receipt item group** based on the combination of retailer and product identifier.
2. Each group is linked to a canonical product through automated matching confirmed by human operators.

This ground truth has been curated over the platform’s operational lifetime and is considered highly reliable for the retailer and time period studied.

4.3.4 4.3.4 Receipt Images

Receipt images are photographs captured by consumers using the mobile application. These images vary significantly in quality due to different lighting conditions, camera angles, and device capabilities. For each receipt in our dataset, the image processing pipeline consists of three stages:

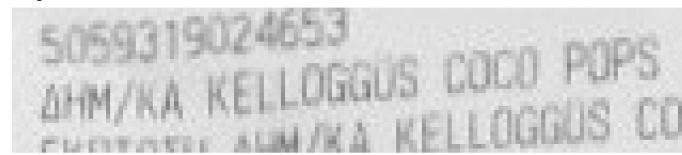
1. **Full receipt retrieval.** The original photograph is downloaded from cloud object storage. Figure 4.4a shows a typical receipt photograph — note the perspective distortion, uneven lighting, and paper creases that are common in user-captured images.
2. **OCR bounding box detection.** Google Vision API’s `documentTextDetection` is applied to each receipt image, producing word-level and paragraph-level bounding boxes with confidence scores. Figure 4.4b illustrates the detected text regions overlaid on the receipt — the API identifies 115 distinct text regions in this example, including both product lines and non-product elements (store header, totals, barcodes).
3. **Line-item cropping.** Each receipt item’s OCR text is matched to its corresponding Vision API paragraph via token overlap. The matched bounding box is used to crop the line-item region with 5 pixels of padding. The resulting crops (Figure 4.4a, right side) capture the visual appearance of individual product lines — character shapes, spacing, alignment, and font characteristics.

Only the cropped regions are retained for the experiments; full receipt images are deleted after processing to protect user privacy.

Figure 4.4: Receipt image processing pipeline. (a) A raw receipt photograph from a Greek supermarket (Sklavenitis). (b) Four line-item crops extracted from the receipt using OCR bounding boxes. Each crop captures a single product line with its barcode, description, and price. Note the uniform visual appearance across crops: identical monospaced font, uppercase Greek text, and fixed column layout — this visual homogeneity explains why image features provide limited discriminative value for product matching.

Figure 4.4b: Google Vision OCR bounding box detection. (a) The original receipt photograph. (b) The same image with detected OCR paragraph regions highlighted as colored

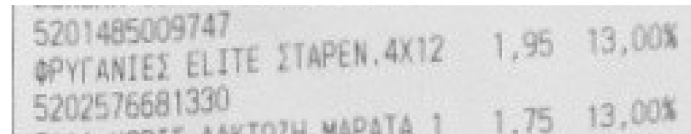
Figure 4.4: Receipt photograph and extracted line-item crops
(a) Raw receipt photograph



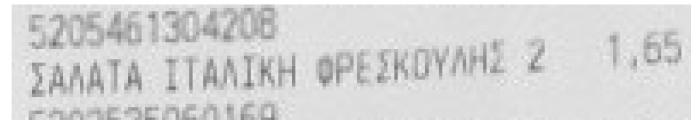
Crop #2: "BONORA CORN FLAKES BIT.& ΣΙΑΗΡΟΣ 375ΓΡ"



Crop #3: "ΦΡΥΓΑΝΙΕΣ ELITE ΣΤΑΠΕΝ.4X12G ΟΙΚΟΓ.ΣΥΣΚ"



Crop #4: "ΣΑΛΑΤΑ ΙΤΑΛΙΚΗ ΦΡΕΣΚΟΥΛΗΣ 200G"



Σχήμα 5: Figure 4.4: Receipt photograph and line-item crops

overlays. Each color represents a distinct paragraph-level text region identified by the Vision API. The bounding box coordinates are used to locate and crop individual line items for the image modality experiments.

4.4 Dataset Statistics

Table 4.1: Dataset statistics

Metric	Count
Total receipt line items	930,490
Unique receipts	36,018
Unique products matched	3,503
Unique brands	570
Unique product subcategories	201
Line items with image crops	22,694
Mean items per receipt	25.8

Metric	Count
Median items per receipt	19.0

Note: The 3,503 unique products are those that appear in the food-category receipt line items during the study period. However, the retrieval evaluation is conducted against the full product catalog of 41,440 entries (spanning all categories and including inactive products), which is the realistic production scenario — the system must search the entire catalog, not just the products known to appear in receipts.

4.4.1 4.3.5 Dataset Examples

To illustrate the matching challenge, we present representative examples from the dataset, showing the OCR text as captured from the receipt alongside the canonical product name from the catalog.

Table 4.2a: Easy matching examples (high lexical overlap)

#	Receipt OCR Text	Catalog Product Name	Brand	Category
1	σκλαβενιτησ σουπιεσ με σπανακι 400g	Σκλαβενίτης Σουπιές με σπανάκι 400g	Σκλαβενίτης	Πρωτεϊνια φαγητά
2	ανανασ εισαγωγησ	Ανανάς εισαγωγής (μαναβική) — τιμή κιλού	Μαναβική	Φρούτα
3	μανιταρια μαρατα κομμενα 230gr	Μανιτάρια MAPATA κομμένα 230gr	Μαράτα	Κονσέρβες
4	Μπανάνες BAJELLA εισαγωγής (μαναβική) - τιμή κιλού	Μπανάνες BAJELLA εισαγωγής (μαναβική) - τιμή κιλού	Bajella	Μπανάνες
5	KOYBEPTOYPA NESTLE DESSERT 170G	Κουβερτούρα NESTLE Dessert 170gr	Nestle Dessert	Κουβερτούρες
6	Spaghettoni BARILLA No7 500gr	Spaghettoni BARILLA No7 500gr	Barilla	Ζυμαρικά μακριά

#	Receipt OCR Text	Catalog Product Name	Brand	Category
7	Ρεβίθια ΣΚΛΑΒΕΝΙΤΗΣ αποφλοιωμένα 500gr	Ρεβίθια ΣΚΛΑΒΕΝΙΤΗΣ αποφλοιωμένα 500gr	Σκλαβενίτης	Ρεβύθια
8	Ρύζι AGRINO Λαΐς καρολίνα 500gr	Ρύζι AGRINO Λαΐς καρολίνα 500gr	Agrino	Ρύζι
9	KANEΛONIA MELISSA 250G	Κανελόνια MELISSA 250gr	Melissa	Ζυμαρικά
10	ΠΙΓΚΑΤΟΝΙ MISKO 500G	Πιγκατόνι MISKO 500gr	Misko	Ζυμαρικά κοντά

In these cases, the OCR text closely matches the product name, differing only in case (uppercase vs. mixed), minor punctuation, or weight format (“G” vs. “gr”). Statistical baselines (TF-IDF) handle these cases well because the high lexical overlap ensures strong character n-gram matches. Examples 4, 6, 7, and 8 are exact matches — these occur when the receipt text is generated directly from the product catalog (e.g., in digital receipts or when the OCR captures the full product name without truncation). Examples 5, 9, and 10 show the more typical case of minor case/format differences that TF-IDF handles effortlessly.

Table 4.2b: Hard matching examples (low or zero lexical overlap)

Figure 4.4b: OCR bounding box detection on receipt image

(a) Original receipt



(b) Google Vision OCR bounding boxes (115 regions)



Σχήμα 6: Figure 4.4b: OCR bounding box detection

#	Receipt OCR Text	Catalog Product Name	Brand	Difficulty
1	μερεντα παυλιδη 1kg	Πραλίνα φουντουκιού MERENDA 1kg	Παυλίδης Merenda	Language mixing: Greek OCR “μερεντα” vs Latin brand “MERENDA”, plus different product description “πραλίνα φουντουκιού”
2	μηλα σταρκιν εγχωρια	Μήλα στάρκιν ελληνικά (μαναβική) — τιμή κιλού	Μαναβική	Dialectal: “εγχωρια” (domestic) vs “ελληνικά” (Greek) — same meaning, zero token overlap
3	blueberries εισαγωγήσ 125g	Blueberries μύρτιλα εισαγωγής 125γρ.	Μαναβική	Missing synonym: catalog adds “μύρτιλα” not in OCR; weight “125g” vs “125γρ”

#	Receipt OCR Text	Catalog Product Name	Brand	Difficulty
4	μερεντα με φουντουκι 360g	Πραλίνα φουντουκιού MERENDA 360gr	Παυλίδης Merenda	Zero lexical overlap — completely different vocabulary for the same product
5	κολοκυθια εγχωρια	Κολοκυθάκια ελληνικά (μαναβική) — τιμή κιλού	Μαναβική	Mixed script (“ia” Latin chars in Greek word), abbreviation, dialectal
6	DIGESTIVE BAR ΠΟΡΤ&ΜΑΥΡΗ ΣΟΚΟΛ 28ΓΡ 5ΤΕΜ	Μπάρες Δημητριακών ΠΑΠΑΔΟΠΟΥΛΟΥ Digestive μαύρη σοκολάτα & πορτοκάλι 5x28gr	Digestive Παπαδο- πούλου	Extreme abbreviation: “ΠΟΡΤ”=“πορτοκάλι”, “ΣΟ- ΚΟΛ”=“σοκολάτα”, “5ΤΕΜ”=“5x”, brand missing
7	ΑΡΤ/ΣΜΑ ΠΑΠΑΔ/ΛΟΥ ΓΕΥΣΗ ΣΙΤΟΥ ΦΟΡΜΑ 700G	Ψωμί τοστ ΠΑΠΑΔΟΠΟΥΛΟΥ γεύση^2 700gr	Παπαδοπούλου Γεύση2	“ΑΡΤ/ΣΜΑ” (αρτοσκεύα- σμα) = “Ψωμί τοστ”; “ΠΑ- ΠΑΔ/ΛΟΥ” = “ΠΑΠΑΔΟ- ΠΟΥΛΟΥ”

#	Receipt OCR Text	Catalog Product Name	Brand	Difficulty
8	APT/ΣΜΑ ΚΡΙΣ ΚΡΙΣ BRIOCHE ΧΑΜΠ 4Τ 360G	Ψωμάκια ΚΡΙΣ ΚΡΙΣ Selection the big brioche burger 360gr	Κρις Κρις	“APT/ΣΜΑ” = “Ψωμάκια”; “ΧΑΜΠ” = “burger”; “4Τ” = unmentioned in catalog
9	ΦΡΑΟΥΛΕΣ ΕΓΧΩΡΙΕΣ.	Φράουλες ελληνικές (μαναβική) – τιμή κιλού	Μαναβική	“ΕΓΧΩΡΙΕΣ” vs “ελληνικές”; catalog adds metadata “(μαναβική) – τιμή κιλού”
10	ΠΑΤΑΤΕΣ ΑΙΓΥΠΤΟΥ.	Πατάτες εισαγωγής (μαναβική) - τιμή κιλού	Μαναβική	Origin generalization: “ΑΙΓΥΠΤΟΥ” (Egypt) → “εισαγωγής” (imported)
11	ΛΕΜΟΝΙΑ ΕΓΧΩΡΙΑ	Λεμόνια ελληνικά (μαναβική) – τιμή κιλού	Μαναβική	Same dialectal pattern as #2, #5, #9 — “ΕΓΧΩΡΙΑ” vs “ελληνικά”
12	ΠΟΡΤΟΚΑΛΙΑ ΜΕΡΛΙΝ ΕΓΧΩΡΙΑ	Πορτοκάλια Μέρλιν ελληνικά χυμού (μαναβική) – τιμή κιλού	Μαναβική	Dialectal + catalog adds “χυμού” (juice variant) not in OCR

These examples illustrate the diverse challenges that make receipt-to-product matching difficult: language mixing between Greek and Latin script (examples 1, 4, 6), dialectal synonyms with zero token overlap (2, 5, 9, 10, 11, 12), extreme POS abbreviations that truncate words to 3-4 characters (6, 7, 8), and mixed-script OCR artifacts where individual characters switch between Greek and Latin (5). The neural models (T1, T2) succeed on these cases by learning the semantic equivalences (e.g., “μερεντα” = “MERENDA” = “πραλίνα φουντουκιού”, “εγχωρια” = “ελληνικά” = “εισαγωγή”) through exposure to thousands of receipt-product training pairs. Statistical baselines have no mechanism to bridge these vocabulary gaps.

Table 4.2c: Product catalog disambiguation challenge — near-identical entries

The product catalog contains many entries that are nearly identical, differing only in a single attribute such as sub-brand, variant, weight, or flavor. This table illustrates the disambiguation challenge across three product families:

Panel A: Bread products (Ψωμί τoστ/φέτες) — same brand, different sub-brands and variants

#	Product Name	Brand	Distinguishing Attribute
1	Ψωμί ΠΑΠΑΔΟΠΟΥΛΟΥ Χωριανό πολυσπορο 540gr	Παπαδοπούλου Χωριανό	Χωριανό, πολυσπορο, 540gr
2	Ψωμί ΠΑΠΑΔΟΠΟΥΛΟΥ Χωριανό σταρένιο με 6 δημητριακά 500gr	Παπαδοπούλου Χωριανό	Χωριανό, σταρένιο, 500gr
3	Ψωμί ΠΑΠΑΔΟΠΟΥΛΟΥ Αρχαία Σπορά σύμμικτο σταρένιο 500gr	Παπαδοπούλου Αρχαία Σπορά	Αρχαία Σπορά, σύμμικτο

#	Product Name	Brand	Distinguishing Attribute
4	Ψωμί ΠΑΠΑΔΟΠΟΥΛΟΥ Αρχαία Σπορά ολικής άλεσης 500gr	Παπαδοπούλου Αρχαία Σπορά	Αρχαία Σπορά, ολικής
5	Ψωμί τοστ ΠΑΠΑΔΟΠΟΥΛΟΥ γεύση^2 700gr	Παπαδοπούλου Γεύση2	Γεύση2, 700gr
6	Ψίχα τοστ ΚΑΡΑΜΟΛΕΓΚΟΣ σταρένιο 480gr	Καραμολέγκος ψίχατοστ	Different brand, “ψίχα” not “ψωμί”

When the receipt reads “ΨΩΜΙ ΠΑΠΑΔΟΠ 500G”, the model must distinguish between entries 2, 3, and 4 — all 500gr Παπαδοπούλου bread products. The discriminating tokens (“Χωριανό”, “Αρχαία Σπορά”, “σύμμικτο”, “ολικής”) may be truncated or absent in the OCR text.

Panel B: Cereal products (Δημητριακά) — many brands and variants in the same category

#	Product Name	Brand	Distinguishing Attribute
1	Δημητριακά KELLOGG’S Corn Flakes original 375gr	Corn Flakes	Corn Flakes, original
2	Δημητριακά KELLOGG’S ALL BRAN Flakes 375gr	All Bran	All Bran, same weight
3	Δημητριακά KELLOGG’S FROSTIES 250gr	Kellogg’s Frosties	Frosties, different weight
4	Δημητριακά QUAKER νιφάδες βρώμης 500gr	Quaker oats	Different brand entirely

#	Product Name	Brand	Distinguishing Attribute
5	Πίτυρο FYTRO βρώμης μούσλι 500gr	Fytro	“Πίτυρο” not “Δημητριακά”

The receipt text “ΔΗΜΗΤΡ ΚΕΛΛΟΓΓ’S 375G” matches both entries 1 and 2 with nearly identical character overlap. The model must learn that the brand sub-name (Corn Flakes vs. All Bran) is the critical distinguishing feature.

Panel C: Pasta products (Ζυμαρικά) — shape, brand, and size variations

#	Product Name	Brand	Distinguishing Attribute
1	Μακαρόνια MELISSA No10 500gr	Melissa	Shape No10
2	Μακαρόνια MAKBEA No2 500gr	Μακβέλ	Different brand, No2
3	Πέννες ριγέ MAKBEA 500gr	Μακβέλ	Different shape (πέννες)
4	Ριγκατόνι MISKO 500gr	Misko	Different brand and shape
5	Spaghettoni BARILLA No7 500gr	Barilla	Latin name, No7
6	Κανελόνια MELISSA 250gr	Melissa	Different shape and weight

All six products are pasta in similar weight packaging. The receipt abbreviation “ΜΑΚΑΡ ΜΕΛΙΣΣΑ 500G” must be resolved to entry 1 and not confused with entry 6 (same brand, different shape and weight) or entry 2 (same shape description, different brand).

Table 4.2d: Sample receipt walkthrough — from OCR to product matching

To provide a concrete view of the matching task as encountered in practice, we present a representative food receipt with 12 line items. Each row shows the raw OCR text as captured from the receipt image, the ground-truth product match from the catalog, and the matching difficulty level.

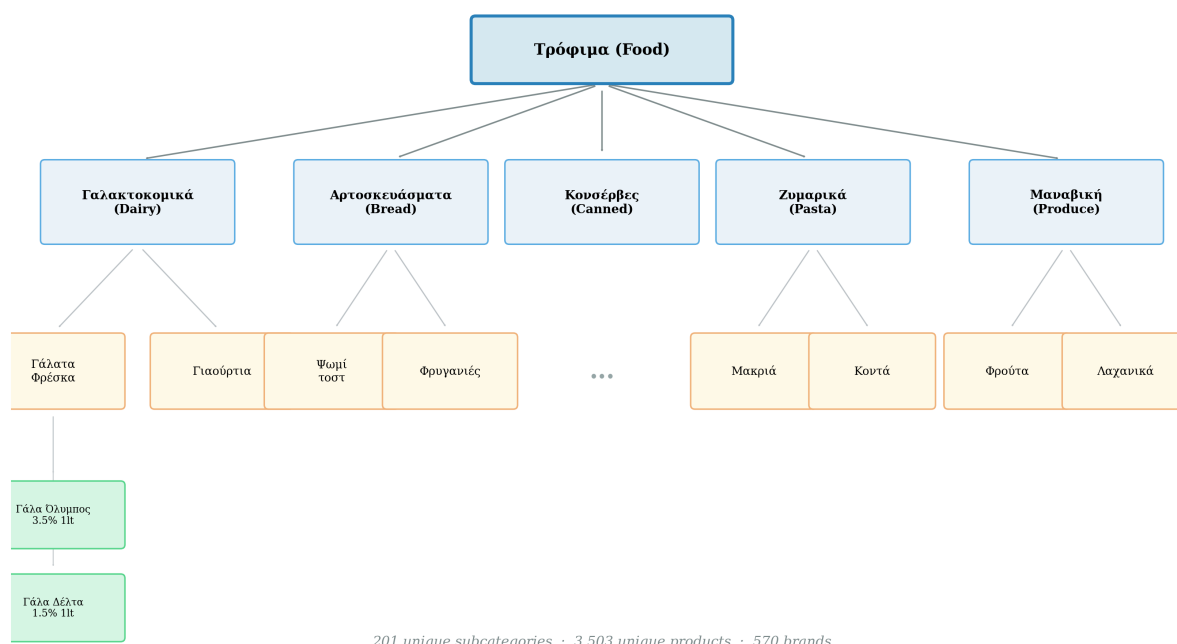
#	Receipt OCR Text	Matched Catalog Product	Brand	Price	Difficulty
1	TOMATA ΤΡΙΜΜΕΝΗ ΚΥΚΝΟΣ 680ΓΡ ΓΥΑΛ	Τομάτα ΚΥΚΝΟΣ τριμμένη 680gr	Kyknos	1.57€	Medium: “ΚΥ- ΚΝΟΣ”→“ΚΥΚΝΟΣ”, “ΓΥΑΛ” (γυάλινο) absent from catalog
2	TOMATA ΚΥΚΝΟΣ ΠΕΡΑΣΤ.ΣΕ ΜΥΛ.ΓΥΑΛ700	Τομάτα ΚΥΚΝΟΣ περαστή στο μύλο 700gr	Kyknos	1.60€	Hard: “ΠΕ- ΡΑΣΤ.”→“περαστή”, “ΜΥΛ.”→“μύλο”, “ΓΥΑΛ700”→“700gr”
3	ΚΡΕΜΑ ΠΑΝ Χ.ΓΛ ΒΑΛΣΑΜΙΚΗ 500ML	Κρέμα βαλσάμικο ΠΑΝ 500ml	Παν	4.44€	Medium: word reordering, “Χ.ΓΛ” (χωρίς γλουτένη) absent
4	ΚΑΝΕΛΟΝΙΑ MELISSA 250G	Κανελόνια MELISSA 250gr	Melissa	1.17€	Easy: near-exact match
5	ΜΑΚΑΡΟΝΙΑ ΝΟ 10 MELISSA 500ΓΡ	Μακαρόνια MELISSA No10 500gr	Melissa	2.25€	Easy: case/spacing differences only
6	ΜΑΚΑΡΟΝΙΑ ΜΑΚΒΕΛ Νο2 500ΓΡ.	Μακαρόνια ΜΑΚΒΕΛ No2 500gr	Μακβέλ	1.24€	Easy: near-exact match

#	Receipt OCR Text	Matched Catalog Product	Brand	Price	Difficulty
7	ΠΕΝΕΣ ΜΑΚΒΕΛ ΡΙΓΕ 500G	Πέννες ριγέ ΜΑΚΒΕΛ 500gr	Μακβέλ	0.90€	Medium: “ΠΕ- ΝΕΣ”→“Πέννες” (missing double ν), word reorder
8	ΜΠΟΥΓΑΤΣΑ ΚΡΕΜΑ ΜΑΡΑΤΑ 450ΓΡ	Μπουγάτσα ΜΑΡΑΤΑ με κρέμα 450gr	Μαράτα	1.90€	Medium: word reordering (“ΚΡΕΜΑ ΜΑΡΑΤΑ” → “ΜΑΡΑΤΑ με κρέμα”)
9	ΡΙΓΚΑΤΟΝΙ ΜΙΣΚΟ 500G	Ριγκατόνι ΜΙΣΚΟ 500gr	Misko	1.02€	Easy: case/format only
10	ΤΟΜΑΤΟΠΟΛΤΟΣ 28% ΚΥΚΝΟΣ 410ΓΡ	Τοματοπολτός ΚΥΚΝΟΣ 28% μεταλλικό 410gr	Kyknos	1.95€	Medium: “ΚΥ- ΚΝΟΣ”→“ΚΥΚΝΟΣ”, “μεταλλικό” absent
11	ΜΑΡΜΕΛΑΔΑ Χ.ΓΛ.Η.ΗΑΒΙΤΣ ΦΡΑΟΥΛ.280G	Μαρμαλάδα HEALTHY HABITS φράουλα 280gr	Healthy Habits	3.55€	Hard: “Η.ΗΑΒΙΤΣ”→“HEALTHY HABITS”, “ΦΡΑ- ΟΥΛ.”→“φράουλα”

#	Receipt OCR Text	Matched Catalog Product	Brand	Price	Difficulty
12	APT/ΣΜΑ ΠΑΠΑΔ/ΛΟΥ ΓΕΥΣΗ ΣΙΤΟΥ ΦΟΡΜΑ 700G	Ψωμί τoστ ΠΑΠΑΔΟΠΟΥΛΟΥ γεύση^2 700gr	Παπαδοπούλου Γεύση2	2,30€	Very hard: “APT/ΣΜΑ”→“Ψωμί τοστ”, “ΠΑ- ΠΑΔ/ΛΟΥ”→“ΠΑΠΑΔΟΠΟ

This receipt illustrates the range of matching difficulty within a single shopping trip. Items 4, 5, 6, and 9 are straightforward matches that any method can resolve. Items 1, 3, 7, 8, and 10 require handling abbreviations (“ΓΥΑΛ” for “γυάλινο”, “Χ.ΓΛ” for “χωρίς γλουτένη”), word reordering, and Greek-to-Latin brand transliteration (“ΚΥΚΝΟΣ”→“KYKNOS”). Items 2, 11, and 12 are the most challenging: extreme abbreviation with slash notation (“APT/ΣΜΑ”, “ΠΑΠΑΔ/ΛΟΥ”), period-truncated words (“ΠΕΡΑΣΤ.”, “ΦΡΑΟΥΛ.”), and brand abbreviations (“H.HABITS”). These hard cases are where neural models provide the most value over statistical baselines.

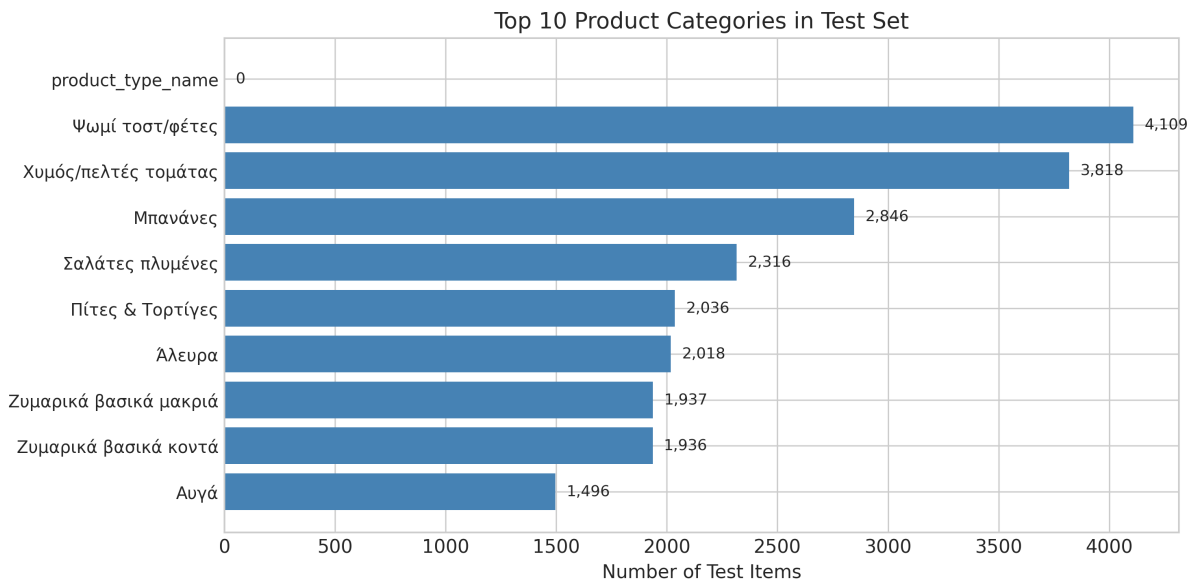
Note: This receipt also contained non-food items (cleaning products, personal care) that are filtered out of our dataset. The 12 items shown are those that belong to the food category and have confirmed ground-truth product matches.



Σχήμα 7: Figure 4.3: Product category tree

Figure 4.3: Product category tree (food subtree excerpt). The food category is organized hierarchically: broad categories (Dairy, Bread, Pasta, Produce) subdivide into subcategories (e.g., Fresh Milk, Yogurt under Dairy), which contain individual products. The dataset spans 201 unique subcategories, 3,503 unique products, and 570 brands. Products within the same subcategory (e.g., different milk brands) create the hardest disambiguation challenges for the matching system.

Table 4.2e: Product category distribution in the food subtree (top 10)



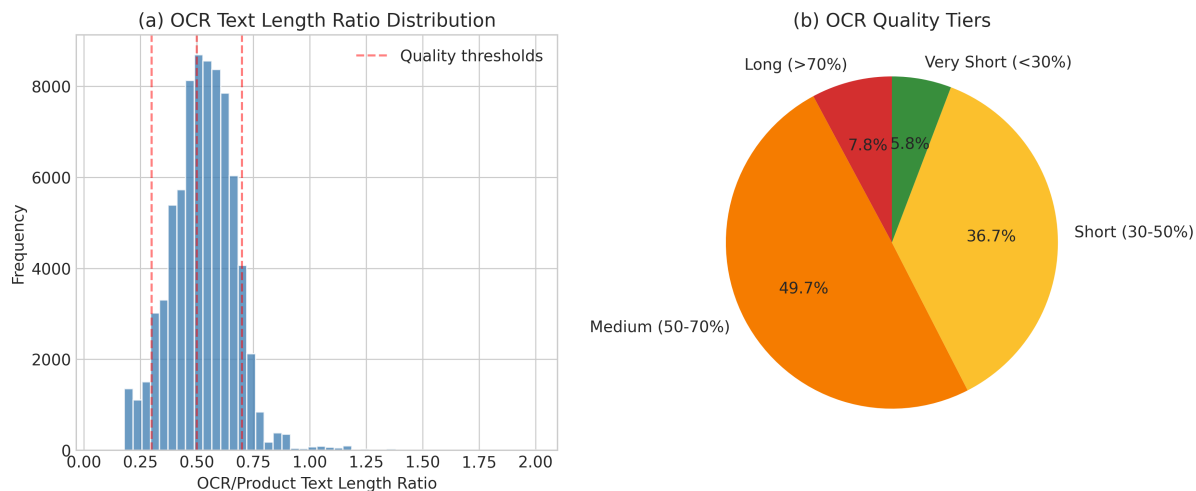
Σχήμα 8: Figure 4.1: Product category distribution

Figure 4.1: Distribution of receipt items per product category in the food subtree. The long-tail distribution shows that a small number of popular categories (bread, tomato products, bananas) account for a disproportionate share of the dataset, while a large number of niche categories contain relatively few items. This distribution reflects real-world purchasing patterns and creates an imbalanced training signal that the contrastive learning framework must handle.

Table 4.3: OCR text length statistics

Statistic	Characters	Tokens (whitespace-split)
Mean	34.3	5.3
Median	34.0	5.0
Min	1	1
Max	235	43

Statistic	Characters	Tokens (whitespace-split)
Std. Dev.	10.0	1.8



Σχήμα 9: Figure 4.2: OCR text length distribution

Figure 4.2: Distribution of OCR text lengths in the dataset. The histogram shows the character-level length distribution of receipt OCR text. The majority of items fall in the 20–45 character range, with a median of 34 characters. The distribution is right-skewed, with a small number of items exceeding 100 characters — typically multi-line receipt entries or items with extended product descriptions.

4.5 4.5 Data Anonymization

To ensure compliance with data protection regulations and ethical research practices, a comprehensive anonymization protocol was applied:

1. **Identifier replacement:** All internal IDs (receipt, line item, product, brand, group) are replaced with sequential integers starting from 1. The mapping is one-way and not stored.
2. **Retailer identification:** The retailer is a major Greek supermarket chain. While private-label product names in the dataset examples make the retailer identifiable, no commercially sensitive information (sales volumes, pricing strategies, market share) is disclosed. The data is used solely for academic research purposes.
3. **User data exclusion:** No user-identifiable information is exported — no user IDs, emails, phone numbers, loyalty card numbers, or demographic data.

4. **Receipt metadata exclusion:** Receipt signatures (containing tax identification numbers), store locations, and timestamps beyond the month are excluded.
5. **Image privacy:** Only line-item crops are retained. Full receipt images, which may contain loyalty card numbers, payment details, or store information, are deleted after cropping.
6. **Data storage:** Anonymized data files are stored only on the researcher’s local machine and a private Google Drive folder. They are not committed to the code repository or shared publicly.

The source of the data is described in the thesis as: “Data provided by a Greek retail technology company, used with permission for academic research purposes.”

4.6 4.6 Data Preprocessing

4.6.1 4.6.1 Greek Text Normalization

Receipt OCR text and product catalog names undergo normalization to reduce surface-level variation. The normalization pipeline, ported from the production system’s PHP implementation, consists of:

1. **Character filtering:** Remove non-alphanumeric characters except spaces, keeping Greek (Unicode range U+0370–U+03FF) and Latin characters.
2. **Accent removal:** Map all accented Greek characters to their unaccented equivalents (e.g., $\alpha \rightarrow \alpha$, $\epsilon \rightarrow \epsilon$, $\eta \rightarrow \eta$, $\iota \rightarrow \iota$, $\omicron \rightarrow \omicron$, $\upsilon \rightarrow \upsilon$, $\omega \rightarrow \omega$).
3. **Final sigma normalization:** Convert terminal sigma ς to standard sigma σ .
4. **Whitespace normalization:** Collapse multiple spaces to single spaces, trim leading/trailing whitespace.
5. **Numeric pattern preservation:** Preserve decimal numbers (e.g., “3.5”) while removing other punctuation, as these often encode product attributes.

The exact normalization implementation is detailed in Appendix A. To illustrate the effect of normalization, Table 4.5 shows before-and-after examples for both receipt OCR text and product catalog entries:

Table 4.5: Text normalization examples

Stage	Receipt OCR Text	Product Catalog Text
Raw input	ΓΑΛΑ ΦΡΕΣΚΟ ΔΕΛΤΑ 1.5LT	Γάλα Φρέσκο Πλήρες 3.5% Όλυμπος 1lt
After cleaning	ΓΑΛΑ ΦΡΕΣΚΟ ΔΕΛΤΑ 1.5LT	Γάλα Φρέσκο Πλήρες 3.5% Όλυμπος 1lt
After lowercasing	γαλα φρεσκο δελτα 1.5lt	γαλα φρεσκο πληρες 3.5% ολυμπος 1lt
After accent removal	γαλα φρεσκο δελτα 1.5lt	γαλα φρεσκο πληρεσ 3.5% ολυμπος 1lt
Final normalized	γαλα φρεσκο δελτα 1.5lt	γαλα φρεσκο πληρεσ 3.5% ολυμπος 1lt

Stage	Receipt OCR Text	Product Catalog Text
Raw input	TYP.ΦΕΤΑ ΠΟΠ ΔΩΔ.400G	Τυρί Φέτα Π.Ο.Π. Δωδώνη 400γρ
After cleaning	TYP ΦΕΤΑ ΠΟΠ ΔΩΔ 400G	Τυρί Φέτα Π Ο Π Δωδώνη 400γρ
After lowercasing	τυρ φετα ποπ δωδ 400g	τυρι φετα π ο π δωδωνη 400γρ
After accent removal	τυρ φετα ποπ δωδ 400g	τυρι φετα π ο π δωδωνη 400γρ
Final normalized	τυρ φετα ποπ δωδ 400g	τυρι φετα π ο π δωδωνη 400γρ

Note how normalization brings the two representations closer together (both lowercased, accents removed, final sigma normalized) but significant differences remain: the receipt text is still abbreviated (“TYP” vs. “τυρι”, “ΔΩΔ” vs. “δωδωνη”) and uses different weight notation (“400G” vs. “400γρ”). These remaining gaps are what the neural matching models must bridge.

For the product catalog, a `search_text` field is constructed by concatenating the normalized product name, brand name, and category name. This gives the matching models maximum textual context:

Product Name	Brand	Category	search_text
Γάλα Φρέσκο Πλήρες 3.5% Ολυμπος 1lt	Ολυμπος	Γάλατα Φρέσκα	γαλα φρεσκο πληρες 3.5% ολυμπος 1lt ολυμπος γαλατα φρεσκα
Πραλίνα φουντουκιού MERENDA 1kg	Παυλίδης Merenda	Πραλίνες	πραλινα φουντουκιου merenda 1kg παυλιδησ merenda πραλινες
Μακαρόνια ΜΑΚΒΕΛ No2 500gr	Μακβέλ	Ζυμαρικά κοντά	μακαρονια μακβελ νο2 500gr μακβελ ζυμαρικα κοντα

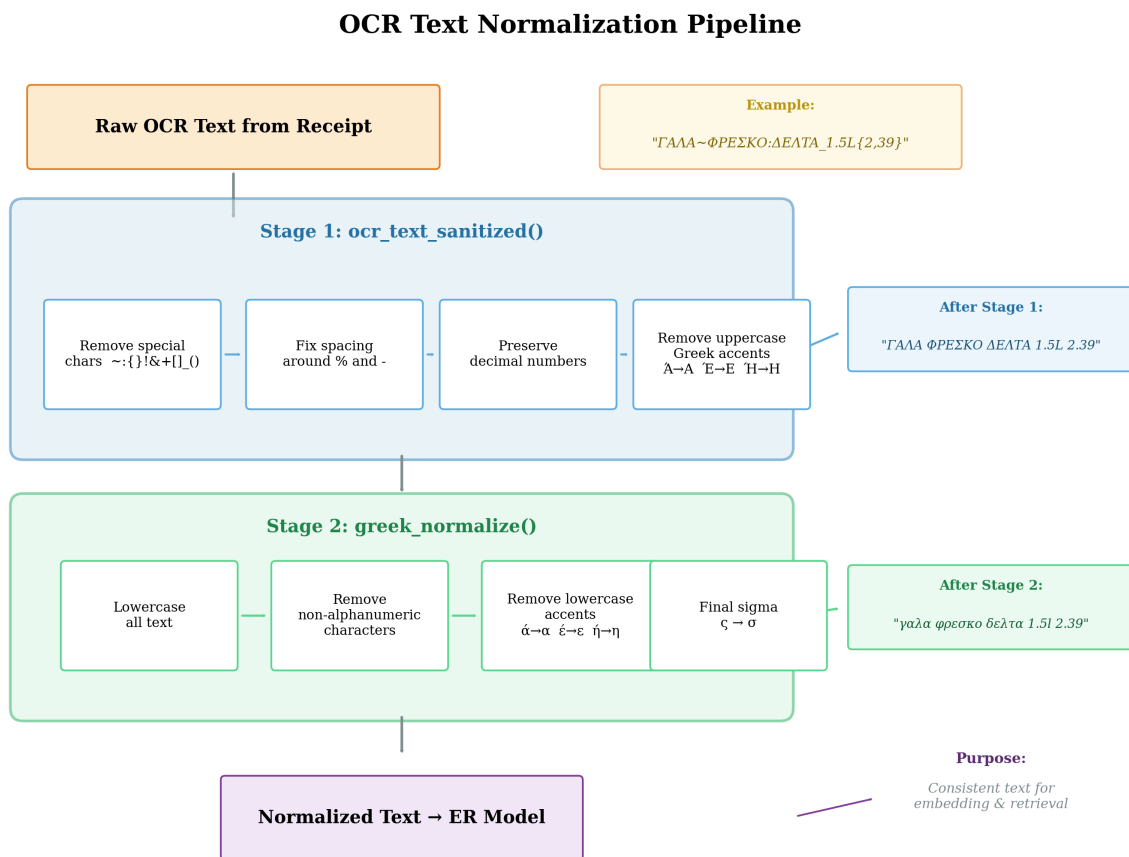
The inclusion of brand and category in the search_text is important: when the receipt reads “MEPENTA ΠΑΥΛΙΔΗ 1KG”, the brand tokens “παυλιδησ merenda” in the search_text provide an additional matching signal, even though the product name itself uses different vocabulary (“πραλινα φουντουκιου” rather than “μερεντα”).

Figure 4.5: Two-stage OCR text normalization pipeline. Raw OCR text from receipts undergoes two sequential processing stages before reaching the entity resolution model. **Stage 1** (ocr_text_sanitized) removes special characters and formatting artifacts (tildes, brackets, underscores), fixes spacing around percentages and negative numbers, preserves decimal number patterns, and strips uppercase Greek accents (e.g., Α→A, Ε→E). **Stage 2** (greek_normalize) applies full normalization: lowercasing, removal of all remaining accents (including lowercase: ᾱ→α, ῑ→ι), and conversion of terminal sigma (ς→σ). Both stages are ported from the production PHP codebase to ensure identical text processing between the existing system and the experimental pipeline. The two-stage design allows the first stage to be used independently when accent-preserving text is needed (e.g., for display), while the second stage produces the fully normalized form used for embedding and retrieval.

4.6.2 Image Preprocessing

Line-item image crops undergo minimal preprocessing:

1. **Resize:** Scale to a fixed height of 224 pixels (ViT/DiT input requirement), preserving aspect ratio.



Σχήμα 10: Figure 4.5: OCR text normalization pipeline

2. **Padding:** Pad to a fixed width with white background.
3. **Normalization:** Apply ImageNet normalization (mean and standard deviation).

No data augmentation is applied to images during preprocessing; augmentation is applied dynamically during training (Section 5.7.5).

4.7 4.7 Data Partitioning

The dataset is split into training, validation, and test sets using a **temporal split** strategy:

Table 4.4: Train / Validation / Test split sizes

Split	Time Period	Line Items	% of Total
Train	Weeks 1–3 (Feb 1–20)	218,024	67.5%
Validation	Feb 21–23	27,435	8.5%
Test	Feb 24–28	77,542	24.0%

The temporal split is chosen over random splitting for two important reasons:

1. **Realism:** In production, a model is trained on historical data and applied to new receipts. A temporal split simulates this scenario, providing a more honest estimate of real-world performance.
2. **Data leakage prevention:** Random splitting could place line items from the same receipt in both training and test sets, inflating performance estimates.

We verify that the product distribution across splits is reasonably balanced and that no systematic drift occurs within the one-month period.

5 Chapter 5: Methodology

5.1 5.1 Problem Formulation

We formulate receipt-to-product matching as a **metric learning retrieval** problem. Let $\mathcal{Q} = \{q_1, \dots, q_N\}$ be the set of receipt line items (queries) and $\mathcal{P} = \{p_1, \dots, p_M\}$ be the product

catalog (gallery). For each query q_i , the task is to rank all products $p_j \in \mathcal{P}$ such that the correct match p_i^* is ranked as high as possible.

Each receipt line item q_i is represented by: - q_i^T : OCR text (string) - q_i^I : Line-item image crop (pixel array) - q_i^N : Numerical attributes (quantity, unit price, final price)

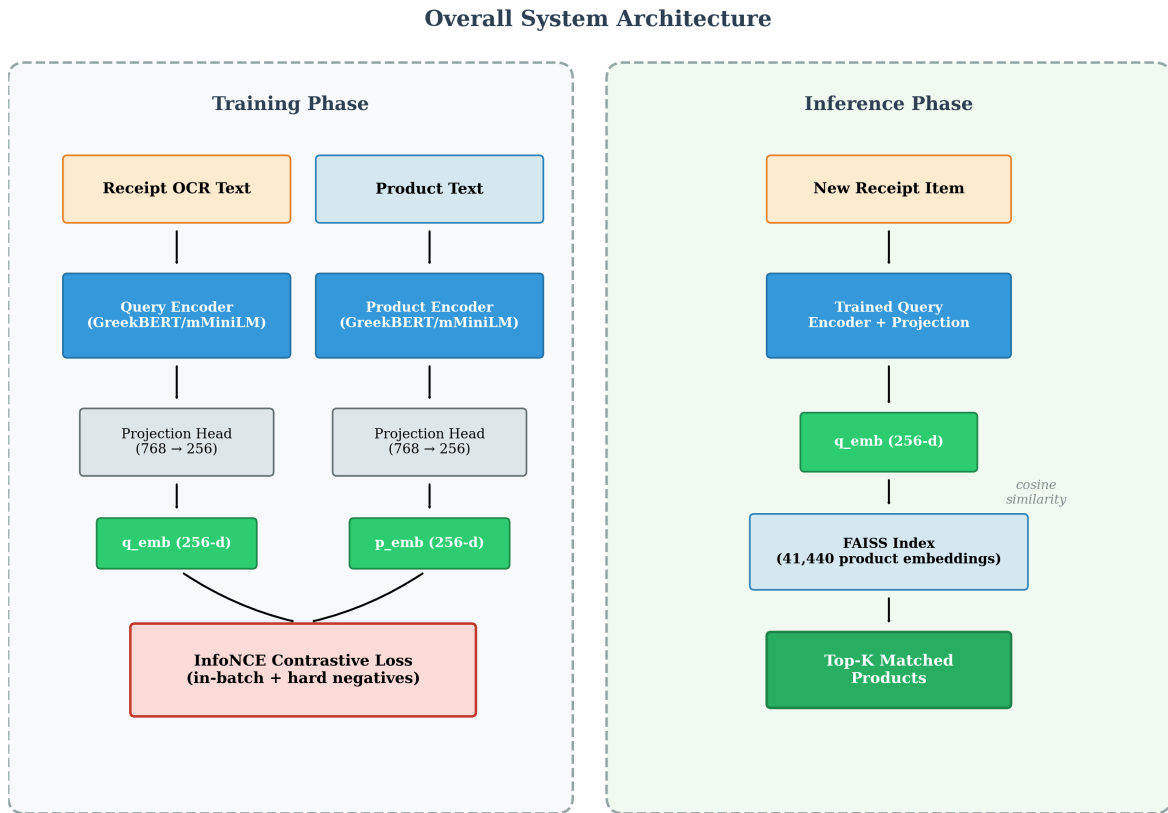
Each product p_j is represented by: - p_j^T : Product name, brand, and category (concatenated string)

The goal is to learn encoding functions $f(\cdot)$ and $g(\cdot)$ such that:

$$\text{sim}(f(q_i), g(p_i^*)) > \text{sim}(f(q_i), g(p_j)) \quad \forall j \neq i^*$$

where $\text{sim}(\cdot, \cdot)$ is cosine similarity in the shared embedding space.

5.2 Overall Architecture



Σχήμα 11: Figure 5.1: Overall system architecture

Figure 5.1: Overall system architecture. During training (left), receipt OCR text and product

catalog text are encoded by separate encoder instances (sharing weights) through projection heads into a 256-dimensional embedding space. InfoNCE contrastive loss trains the system to place matching pairs close together. During inference (right), a new receipt item is encoded and matched against all 41,440 pre-computed product embeddings via FAISS cosine similarity search, returning the top-K candidates.

The proposed framework consists of four main components:

1. **Query Encoder** $f(\cdot)$: Encodes receipt line items (text and/or image) into a fixed-dimensional embedding vector.
2. **Product Encoder** $g(\cdot)$: Encodes catalog products into the same embedding space.
3. **Projection Head**: Maps encoder outputs to a shared embedding space of dimension d .
4. **Contrastive Loss**: Trains the system to place matching pairs close and non-matching pairs far apart.

At inference time, all product embeddings are pre-computed and indexed. For a new receipt line item, the query encoder produces an embedding, and FAISS retrieves the top-K nearest products.

5.3 Experiment Matrix

Ten experimental configurations are evaluated, spanning five categories:

Table 5.1: Experiment matrix — all ten configurations

ID	Category	Text Encoder	Image Encoder	Fusion	Description
B1	Baseline	TF-IDF + cosine	—	—	Statistical text baseline
B2	Baseline	BM25	—	—	Probabilistic text baseline
B3	Commercial API	OpenAI text-embedding-3-small	—	—	General-purpose commercial embeddings
T1	Text-only	GreekBERT	—	—	Greek-specific neural text

ID	Category	Text Encoder	Image Encoder	Fusion	Description
T2	Text-only	mMiniLM	—	—	Lightweight multilingual text
I1	Image-only	—	DiT	—	Visual-only baseline
M1	Multimodal	GreekBERT	DiT	Early (concat)	Feature concatenation
M2	Multimodal	GreekBERT	DiT	Late (score)	Score-level combination
M3	Multimodal	GreekBERT	DiT	Hybrid (cross-attn)	Cross-attention fusion
L1	Multimodal	LayoutLMv3	LayoutLMv3	Native	Integrated model

5.4 5.4 Text Encoders

5.4.1 5.4.1 TF-IDF Baseline (B1)

TF-IDF vectors are computed over the corpus of receipt OCR texts and product names using character n-grams ($n=2,3,4$) to handle abbreviations and subword matches. Cosine similarity between the TF-IDF vectors of a receipt line item and each product name produces the ranking.

5.4.2 5.4.2 BM25 Baseline (B2)

BM25 scoring (Robertson & Zaragoza, 2009) is computed with default parameters ($k_1 = 1.2$, $b = 0.75$). The product catalog is indexed as a document collection, and each receipt OCR text is treated as a query. This baseline approximates the production system’s text matching approach.

5.4.3 5.4.3 GreekBERT (T1)

GreekBERT (Koutsikakis et al., 2020) is a BERT-base model (12 layers, 768 hidden dimension, 12 attention heads, ~110M parameters) pre-trained on Greek text. We use it as the text encoder with mean pooling over the last hidden states:

$$h_T = \text{MeanPool}(\text{GreekBERT}(x_T))$$

The resulting 768-dimensional vector is projected to the shared embedding dimension d via a learned linear projection with ReLU activation:

$$z_T = \text{ReLU}(W_T h_T + b_T), \quad z_T \in \mathbb{R}^d$$

5.4.4 5.4.4 mMiniLM (T2)

The multilingual MiniLM (Wang et al., 2020) is a distilled model (6 layers, 384 hidden dimension, 12 attention heads, ~22M parameters) supporting 100+ languages including Greek. It offers a five-fold reduction in parameters compared to GreekBERT, making it suitable for resource-constrained deployment.

The encoding and projection procedure is identical to GreekBERT, with the hidden dimension adjusted from 768 to 384.

5.4.5 5.4.5 OpenAI text-embedding-3-small (B3)

As an additional baseline, we evaluate OpenAI’s commercial text-embedding-3-small model, a general-purpose multilingual embedding model accessed via API. This model produces 1,536-dimensional embeddings without task-specific fine-tuning. Product texts are embedded using the `search_text` field (concatenation of normalized product name, brand, and category). Receipt OCR texts are embedded directly from the `ocr_text_normalized` field. Matching is performed via cosine similarity between query and product embeddings.

This baseline serves two purposes: (i) it establishes the performance of a state-of-the-art commercial embedding service on our task, and (ii) it quantifies the benefit of domain-specific fine-tuning (T1, T2) over general-purpose embeddings. The total API cost for embedding 77,542 queries and 41,440 products was approximately \$0.04.

5.5 5.5 Image Encoder

5.5.1 5.5.1 DiT (I1, M1-M3)

DiT-base (Li et al., 2022a) is a ViT-base model (12 layers, 768 hidden dimension, 12 attention heads, ~86M parameters) pre-trained on document images. Each line-item crop is split into 16×16 patches, linearly projected, and processed by the transformer encoder.

The [CLS] token representation is used as the image embedding:

$$h_I = \text{DiT}(x_I)_{[\text{CLS}]}$$
$$z_I = \text{ReLU}(W_I h_I + b_I), \quad z_I \in \mathbb{R}^d$$

5.6 5.6 Multimodal Fusion

5.6.1 5.6.1 Early Fusion (M1)

The text embedding h_T and image embedding h_I are concatenated and passed through a two-layer MLP:

$$h_{\text{fused}} = \text{ReLU}(W_2 \cdot \text{ReLU}(W_1 [h_T; h_I] + b_1) + b_2)$$
$$z_{\text{fused}} = \text{Normalize}(W_p h_{\text{fused}} + b_p), \quad z_{\text{fused}} \in \mathbb{R}^d$$

5.6.2 5.6.2 Late Fusion (M2)

The text and image encoders produce independent embeddings z_T and z_I . For a query q and product p :

$$s_{\text{text}} = \cos(z_T^q, z_T^p)$$
$$s_{\text{image}} = \cos(z_I^q, z_I^p)$$
$$s_{\text{fused}} = \alpha \cdot s_{\text{text}} + (1 - \alpha) \cdot s_{\text{image}}$$

The mixing weight $\alpha \in [0, 1]$ is treated as a hyperparameter optimized on the validation set.

The text and image encoders are pre-trained independently (using configurations T1 and I1 respectively) before fusion.

5.6.3 Hybrid Fusion with Cross-Attention (M3)

A cross-attention module allows the text and image representations to interact:

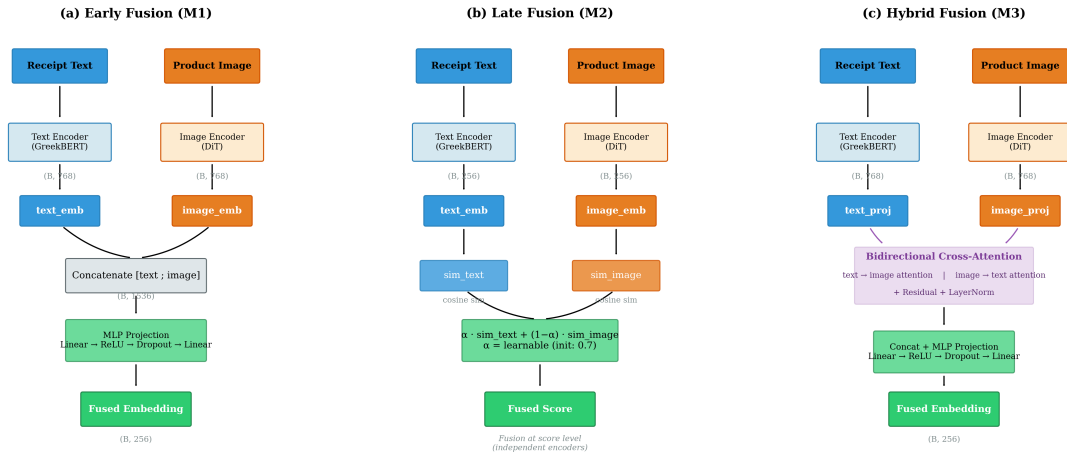
$$h'_T = h_T + \text{MultiHeadCrossAttention}(Q = h_T, K = h_I, V = h_I)$$

$$h'_I = h_I + \text{MultiHeadCrossAttention}(Q = h_I, K = h_T, V = h_T)$$

$$h_{fused} = [h'_T; h'_I]$$

$$z_{fused} = \text{Normalize}(\text{MLP}(h_{fused})), \quad z_{fused} \in \mathbb{R}^d$$

The cross-attention uses 8 heads and a hidden dimension of 768. Residual connections ensure that unimodal information is preserved even if cross-modal attention is uninformative.



Σχήμα 12: Figures 5.2–5.4: Multimodal fusion strategies

Figures 5.2–5.4: Comparative architecture of the three multimodal fusion strategies. (a) Early Fusion (M1): Text and image embeddings are concatenated into a single vector of dimension 1536, then projected through a two-layer MLP (Linear→ReLU→Dropout→Linear) to produce a 256-dimensional fused embedding. This allows cross-modal feature interaction from the earliest stage but requires the model to learn joint representations from scratch. **(b) Late Fusion (M2):** Each modality maintains its own encoder and produces independent embeddings. During retrieval, cosine similarity scores from each modality are combined via a learnable weight α (initialized to 0.7, text-biased), parameterized through a sigmoid to ensure $\alpha \in (0, 1)$. **(c) Hybrid Fusion (M3):** Text and image embeddings are processed by separate encoders and then interact through a bidirectional cross-attention module. The output is concatenated and passed through an MLP projection to produce a 256-dimensional fused embedding.

1). Fusion occurs at the score level, not the feature level, preserving modality independence.

(c) Hybrid Fusion (M3): Both embeddings are projected to a common 768-dimensional space and processed through bidirectional multi-head cross-attention (8 heads): text attends to image features and vice versa. Residual connections and LayerNorm preserve the original modality information. The attended representations are concatenated and projected to produce the final 256-dimensional embedding.

5.6.4 5.6.4 LayoutLMv3 (L1)

LayoutLMv3 (Huang et al., 2022) natively processes text, image, and 2D position information in a single transformer. For each line-item crop:

1. OCR text is tokenized and paired with bounding box coordinates (normalized to $[0, 1000]$).
2. The image is split into patches.
3. Text tokens and image patches are concatenated and processed jointly.

The [CLS] token representation serves as the fused embedding:

$$z_L = \text{Normalize}(W_L \cdot \text{LayoutLMv3}(x_T, x_I, x_{bbox})_{[\text{CLS}]} + b_L)$$

Figure 5.5: LayoutLMv3 integrated architecture (L1). Unlike the separate-encoder fusion approaches (M1–M3), LayoutLMv3 processes text tokens, image patches, and 2D bounding box positions through a single unified transformer encoder. The three modalities are embedded separately and concatenated into a single input sequence, allowing joint self-attention across all modalities. The model is pre-trained with three objectives: masked language modeling (MLM), masked image modeling (MIM), and word-patch alignment. Note: this configuration was not trained due to the image probe results (Section 7.1.3).

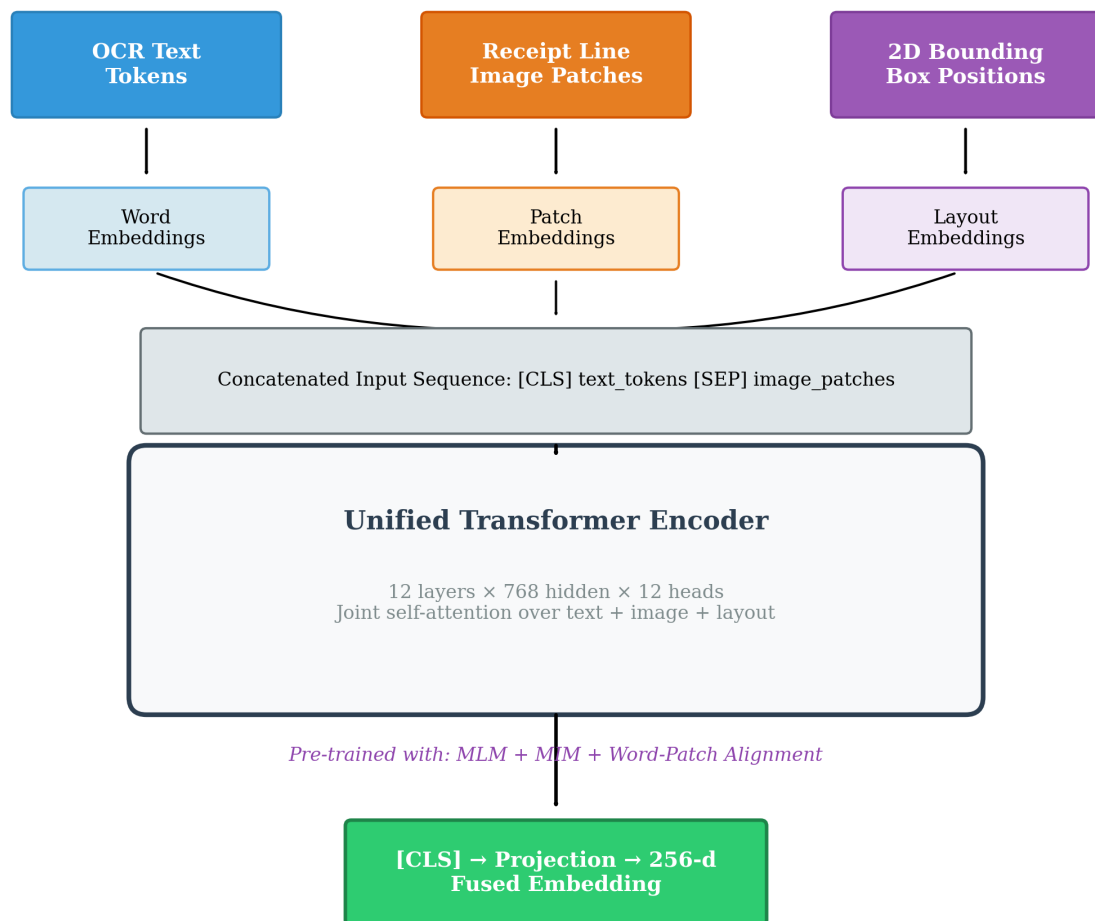
5.7 5.7 Training Procedure

5.7.1 5.7.1 Pair Generation

For each receipt line item in the training set, we generate:

- **One positive pair:** The line item paired with its ground-truth product.
- **Hard negatives:** Products from the same subcategory (e.g., other milks when the correct

LayoutLMv3 Integrated Architecture (L1)



Σχήμα 13: Figure 5.5: LayoutLMv3 integrated architecture

match is a specific milk product). We sample 3 hard negatives per positive.

- **In-batch negatives:** All other products in the mini-batch serve as additional negatives, following the InfoNCE formulation.

5.7.2 5.7.2 Loss Function

We use InfoNCE loss (Oord et al., 2018):

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(\cos(z_i^q, z_i^p) / \tau)}{\sum_{j=1}^N \exp(\cos(z_i^q, z_j^p) / \tau)}$$

where z_i^q is the query embedding, z_i^p is the positive product embedding, and the denominator sums over all products in the batch (including hard negatives). The temperature $\tau = 0.07$ follows common practice (Radford et al., 2021).

5.7.3 5.7.3 Staged Training

For multimodal models (M1-M3), we adopt a staged training approach:

1. **Stage 1 — Text warmup** (5 epochs): Train only the text encoder and projection head. Image encoder weights are frozen.
2. **Stage 2 — Image warmup** (5 epochs): Train only the image encoder and projection head. Text encoder weights are frozen.
3. **Stage 3 — Joint training** (remaining epochs): Train all components jointly with a reduced learning rate.

This staged approach prevents the text modality — which typically converges faster — from dominating the learning signal and causing the image encoder to receive uninformative gradients.

5.7.4 5.7.4 Optimization

- **Optimizer:** AdamW (Loshchilov & Hutter, 2019) with weight decay 0.01.
- **Learning rates:** 2×10^{-4} for pre-trained backbones, 1×10^{-3} for projection heads.
- **Schedule:** Linear warmup (10% of steps) followed by cosine decay.
- **Batch size:** 32 per device, with gradient accumulation over 8 steps for effective batch size of 256.
- **Mixed precision:** FP16 automatic mixed precision (AMP) to reduce memory usage and enable training on a single T4 GPU (16GB VRAM).

- **Early stopping:** Training stops if validation MRR does not improve for 5 consecutive epochs.
- **Maximum epochs:** 20.

5.7.5 5.7.5 Data Augmentation

To improve generalization, the following augmentations are applied dynamically during training:

Text augmentation: - Character dropout: Randomly remove 1-2 characters (simulating OCR errors). - Token shuffle: Randomly swap adjacent tokens with probability 0.1.

Image augmentation: - Random brightness/contrast adjustment ($\pm 10\%$). - Random Gaussian blur ($\sigma \in [0.5, 1.0]$). - Random horizontal shift ($\pm 5\%$ of width).

Augmentations are applied with a combined probability of 0.3 per sample.

5.8 5.8 Evaluation Protocol

5.8.1 5.8.1 Retrieval Setup

For evaluation, we:

1. Encode all products in the catalog to produce a gallery of product embeddings.
2. Index the gallery using FAISS flat (exact) search.
3. For each test line item, compute the query embedding and retrieve the top-K products.
4. Compare the retrieved products to the ground-truth match.

5.8.2 5.8.2 Metrics

We report the following standard retrieval metrics:

Precision@K: Fraction of the top-K retrieved products that are correct:

$$P@K = \frac{|\text{relevant} \cap \text{top-K}|}{K}$$

Since each query has exactly one relevant product, $P@1 = \text{Hit Rate}@1$ and $P@K \leq 1/K$ for $K > 1$.

Recall@K: Whether the correct product appears in the top-K:

$$R@K = \begin{cases} 1 & \text{if correct product} \in \text{top-K} \\ 0 & \text{otherwise} \end{cases}$$

Mean Reciprocal Rank (MRR): The mean of the reciprocal rank of the correct product:

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^N \frac{1}{\text{rank}_i}$$

Hit Rate@K: Fraction of queries where the correct product appears in the top-K results:

$$\text{HR@K} = \frac{1}{N} \sum_{i=1}^N \mathbb{1}[\text{rank}_i \leq K]$$

We report all metrics at $K \in \{1, 5, 10\}$.

6 Chapter 6: Experimental Setup

6.1 6.1 Implementation

The experimental framework is implemented in Python 3.10 using the following key libraries:

- **PyTorch 2.1** (Paszke et al., 2019): Deep learning framework for model implementation and training.
- **HuggingFace Transformers 4.35** (Wolf et al., 2020): Pre-trained model loading and tokenization.
- **FAISS-CPU 1.7** (Johnson et al., 2019): Efficient nearest-neighbor search for evaluation.
- **OmegaConf 2.3**: Hierarchical configuration management.
- **scikit-learn 1.3** (Pedregosa et al., 2011): TF-IDF vectorization and baseline utilities.
- **rank-bm25 0.2**: BM25 implementation for the B2 baseline.

All code is organized in a modular structure with separate components for data processing, model definitions, training loops, and evaluation. Configuration is managed through YAML

files, enabling reproducible experiment execution.

6.2 6.2 Pre-trained Models

Table 6.2: Pre-trained model specifications

Model	HuggingFace ID	Parameters	Hidden Dim	Layers	Heads
GreekBERT	hpaueb/bert-base-greek-uncased-v1	~110M	768	12	12
mMiniLM	microsoft/Multilingual-2021-MiniLM-L12-H384	~22M	384	12	12
DiT-base	microsoft/dit-base	~86M	768	12	12
LayoutLMv3	microsoft/layoutlmv3-base	~133M	768	12	12

All models are loaded with pre-trained weights and fine-tuned during training. The final classification heads of the original models are removed and replaced with our projection heads.

6.3 6.3 Hyperparameters

Table 6.1: Hyperparameter settings for all models

Hyperparameter	Value	Notes
Shared embedding dimension d	256	Ablated over {128, 256, 512}
Temperature τ	0.07	Ablated over {0.03, 0.05, 0.07, 0.1, 0.2}
Backbone learning rate	2×10^{-4}	With linear warmup + cosine decay
Projection head learning rate	1×10^{-3}	With linear warmup + cosine decay

Hyperparameter	Value	Notes
Weight decay	0.01	AdamW
Batch size (per device)	32	
Gradient accumulation steps	8	Effective batch = 256
Maximum epochs	20	
Early stopping patience	5	Based on validation MRR
Warmup ratio	0.1	10% of total training steps
Hard negatives per positive	3	From same subcategory
Late fusion weight α (M2)	Tuned	Grid search over {0.5, 0.6, 0.7, 0.8, 0.9}
Cross-attention heads (M3)	8	
Text augmentation probability	0.3	
Image augmentation probability	0.3	
Random seed	42	Fixed for reproducibility

6.4 Hardware and Computing Environment

All experiments are conducted on **Google Colab Free** with the following specifications:

Resource	Specification
GPU	NVIDIA Tesla T4 (16 GB VRAM)
System RAM	~12 GB
Disk	~100 GB (session-ephemeral)
Runtime	Up to 12 hours per session

FP16 mixed precision training is essential to fit the models within the T4’s 16GB VRAM.

The largest model (LayoutLMv3, ~133M parameters) requires approximately 10GB of GPU memory during training with a batch size of 32.

Estimated training times per model:

Model	Time per Epoch	Total Training	Notes
B1 (TF-IDF)	< 1 min	< 1 min	CPU only
B2 (BM25)	< 1 min	< 1 min	CPU only
B3 (OpenAI)	N/A	~5 min	API calls, no training
T1 (GreekBERT)	~13 min	~195 min (15 epochs)	Best at epoch 13
T2 (mMiniLM)	~10 min	~90 min (9 epochs)	Best at epoch 8, early stop
I1 (DiT)	—	Not conducted	Image probe showed no benefit
M1-M3 (Multimodal)	—	Not conducted	Image probe showed no benefit
L1 (LayoutLMv3)	—	Not conducted	Image probe showed no benefit

6.5 6.5 Reproducibility

To ensure reproducible results:

1. **Random seeds:** All random number generators (Python, NumPy, PyTorch, CUDA) are seeded with the same value (42).
2. **Deterministic operations:** PyTorch deterministic mode is enabled where possible (`torch.use_deterministic_algorithms`).
3. **Configuration versioning:** All experiment configurations are stored in version-controlled YAML files.

4. **Dependency pinning:** Exact library versions are specified in `pyproject.toml`.

Note that perfect reproducibility on GPU is not guaranteed due to non-deterministic CUDA operations in some PyTorch functions, but results should be consistent within $\pm 0.5\%$ across runs.

6.6 6.6 Evaluation Procedure

Evaluation follows this protocol:

1. **Model selection:** For each experiment, the model checkpoint with the highest validation MRR is selected.
2. **Product index construction:** All catalog products are encoded using the product encoder to build the FAISS index.
3. **Test set inference:** Each test line item is encoded and queried against the product index.
4. **Metric computation:** Precision@K, Recall@K, MRR, and Hit Rate@K are computed at $K = 1, 5, 10$.
5. **Statistical significance:** Bootstrap confidence intervals (95%) are computed over 1000 resamples of the test set.
6. **Per-category analysis:** Metrics are also computed per product subcategory to identify category-specific patterns.

6.7 6.7 Step-by-Step Experiment Procedure

To provide a concrete understanding of how each experiment is conducted, we describe the end-to-end process for each model category:

6.7.1 6.7.1 Statistical Baselines (B1, B2)

B1 (TF-IDF): No training is required. The process is: 1. All product `search_text` fields and all receipt `ocr_text_normalized` fields are transformed into character n-gram vectors ($n=2,3,4$) using TF-IDF weighting. 2. For each test receipt item, cosine similarity is computed against all 41,440 product vectors. 3. Products are ranked by similarity; the top-K are returned as predictions.

B2 (BM25): Similar to B1, but uses word-level BM25 scoring instead of character n-gram TF-IDF. The BM25 parameters ($k_1 = 1.2$, $b = 0.75$) are the standard values from Robertson &

Zaragoza (2009).

6.7.2 6.7.2 Commercial API Baseline (B3)

B3 (OpenAI text-embedding-3-small): No training is required. The process is: 1. All product search_text fields (41,440) are sent to the OpenAI Embeddings API, producing 1,536-dimensional vectors. 2. All test receipt ocr_text_normalized fields (77,542) are sent to the same API. 3. Vectors are L2-normalized and cosine similarity is computed using batched matrix multiplication. 4. Products are ranked by similarity for each query.

Total API cost: approximately \$0.04. No fine-tuning or domain adaptation is applied — the model is used as-is, representing the “out-of-the-box” performance of a commercial embedding service.

6.7.3 6.7.3 Neural Text Models (T1, T2)

Training procedure (identical for T1 and T2, differing only in the pre-trained backbone): 1.

Data preparation: 218,024 training pairs are loaded. Each pair consists of (ocr_text_normalized, search_text) — the receipt text and its matching product text. 2. **Tokenization:** Both texts are tokenized using the model’s tokenizer (GreekBERT or mMiniLM) with max length 64, padding, and truncation. 3. **Forward pass:** Both texts are encoded through the shared backbone + projection head to produce 256-dimensional embeddings. 4. **Loss computation:** InfoNCE loss computes a 64×64 similarity matrix for each batch. The diagonal elements (correct pairs) should be high; off-diagonal (in-batch negatives) should be low. 5. **Optimization:** AdamW optimizer with separate learning rates (2e-5 for backbone, 1e-3 for projection head) and OneCycleLR scheduler. 6. **Validation:** After each epoch, the full catalog (41,440 products) is encoded and indexed with FAISS. All 27,435 validation queries are evaluated using MRR, HR@1, HR@5, HR@10. 7. **Checkpointing:** The model with the highest validation MRR is saved. Training stops if MRR doesn’t improve for 5 epochs.

Evaluation procedure: 1. The best checkpoint is loaded. 2. All 41,440 catalog products are encoded → FAISS inner-product index. 3. All 77,542 test receipt items are encoded. 4. For each query, the 10 nearest product embeddings are retrieved. 5. Metrics are computed by checking if the ground-truth product appears in the top-K.

Concrete example of matching flow:

Input: "μερεντα παυλιδη 1kg"

↓ GreekBERT tokenizer

Tokens: [CLS] μερ ##εντα παυλ ##ιδη 1 ##kg [SEP]

↓ GreekBERT encoder (768-dim)

↓ Projection head (768 → 256)

Output: [0.23, -0.15, 0.87, ..., 0.42] (256 numbers)

↓ cosine similarity vs all 41,440 products

Top-1: "πραλινα φουντουκιου merenda 1kg παυλιδη merenda πραλινες"

→ CORRECT MATCH ✓

6.7.4 6.7.4 What the System Learns from Training Data

To provide intuition for why contrastive fine-tuning is so effective, we examine what specific patterns the neural models learn from the training data. The 218,024 training pairs expose the model to recurring receipt-product correspondences, enabling it to learn several types of implicit knowledge:

Abbreviation mappings. Greek POS systems impose strict character limits, producing systematic abbreviations. Through repeated exposure, the model learns these abbreviation conventions:

Abbreviation Pattern	Receipt Text	Product Text	Frequency
Truncation to first syllable	ΓΑΛ	Γάλα (milk)	Very common
	ΤΥΡ	Τυρί (cheese)	Very common
	ΣΟΚΟΛ	Σοκολάτα (chocolate)	Common
Slash abbreviation	ΑΡΤ/ΣΜΑ	Αρτοσκεύασμα → Ψωμί (bread)	Common
	ΠΑΠΑΔ/ΛΟΥ	ΠΑΠΑΔΟΠΟΥΛΟΥ	Common
	Κ/Ψ	Κατεψυγμένο (frozen)	Common
Period truncation	ΦΡΑΟΥΛ.	Φράουλα (strawberry)	Common
	ΠΕΡΑΣΤ.	Περαστή (strained)	Moderate

Abbreviation Pattern	Receipt Text	Product Text	Frequency
Number codes	No10, No2, No7	Pasta shape numbers	Domain-specific
Quantity packing	5TEM, 4T, 3X500G	5τμχ, 4 pieces, 3x500gr	Very common

The model learns these mappings implicitly through the contrastive loss: abbreviated texts are pushed close to their full product names in embedding space. No explicit abbreviation dictionary is provided.

Brand transliteration. Many Greek products have brands written in either Latin or Greek script, and receipts may use either form depending on the POS system:

Receipt Form	Catalog Form	Implicit Learning
μερεντα (Greek)	MERENDA (Latin)	Greek-to-Latin script equivalence
ΚΥΚΝΟΣ (Greek)	KYKNOS (Latin)	Same brand, different alphabet
νεστλε (Greek phonetic)	NESTLE (Latin)	Phonetic transliteration
γιωτης (Greek)	ΓΙΩΤΗΣ (Greek uppercase)	Case normalization

The contrastive training teaches the model that “μερεντα παυλιδη 1kg” and “πραλινα φουντουκιου MERENDA 1kg” refer to the same product, even though the two texts share almost no tokens. This cross-script matching capability is impossible for lexical methods like TF-IDF.

Dialectal and synonym mappings. The receipt and catalog often use different words for the same concept:

Receipt Term	Catalog Term	Meaning
εγχωρια / εγχωριες	ελληνικά / ελληνικές	domestic / Greek
ΑΙΓΥΠΤΟΥ	εισαγωγής	from Egypt / imported
XAMΠ	burger	hamburger abbreviation
X.ΓΛ	χωρίς γλουτένη	gluten-free

Receipt Term	Catalog Term	Meaning
ΜΩΡ/ΛΑ	μωρομάντηλα	baby wipes

These synonym pairs are learned from training examples where the same product appears with different descriptions across multiple receipts. The model builds an internal representation where “εγχωρία” and “ελληνικά” are close in embedding space.

Product-attribute associations. The model also learns to associate product attributes (weight, fat percentage, pack size) with specific products:

- “1LT 3.5%” → likely a full-fat milk product (1 liter)
- “6X330ML” → likely a beer or soft drink multipack
- “5X28GR” → likely a cereal/snack bar multipack
- “500ΓΡ” with “ΜΑΚΑΡ” → likely pasta (standard Greek pasta packaging)

These associations help disambiguate when the product name itself is abbreviated or ambiguous. For instance, “ΓΑΛ 1LT 3.5%” narrows the search space to full-fat 1-liter milks, allowing the model to focus on distinguishing the brand.

Concrete example — tokenization and matching flow for a hard case:

Receipt OCR: "ΑΡΤ/ΣΜΑ ΠΑΠΑΔ/ΛΟΥ ΓΕΥΣΗ ΣΙΤΟΥ ΦΟΡΜΑ 700G"

↓ text normalization

Normalized: "αρτ σμα παπαδ λου γευση σιτου φορμα 700g"

↓ GreekBERT tokenizer (WordPiece)

Tokens: [CLS] αρ ##τ σ ##μα παπ ##αδ λ ##ου γε ##υ ##ση σι ##του
φ ##ορ ##μα 700 ##g [SEP]

↓ GreekBERT encoder → mean pooling → projection

Embedding: 256-dimensional vector

Product text: "ψωμι τοστ παπαδοπουλου γευση 2 700gr παπαδοπουλου γευση2
ψωμι τοστ φετες"

↓ same tokenization and encoding pipeline

Embedding: 256-dimensional vector

Cosine similarity: 0.87 → RANK 1 (correct match!)

Despite sharing almost no full tokens (only “700” is common), the model correctly matches these texts because the contrastive training has taught it that the subword patterns “παπ##αδ” and “παπ##αδ##ο##πουλ##ου” are closely related, and that “αρ##τ σ##μα” in receipt context maps to “ψωμ##ι τοστ” in catalog context.

6.7.5 6.7.5 Image Probe

The image probe tests whether adding visual features to the text model improves accuracy: 1.

Image acquisition: 5,578 receipt photographs were downloaded from the production CDN.

Google Vision API was re-run on each image (~\$8 total cost) to obtain word-level bounding

boxes. 2. **Cropping:** For each receipt item, the corresponding line in the receipt image was

identified by matching OCR text to Vision API words ($\geq 30\%$ token overlap). The bounding

box was used to crop the line region, producing 22,694 line-item images (average size 181×32

pixels). 3. **Mini-split:** Since images were only available for the test period (Feb 24–28), the data

was split into mini-train (Feb 24–26: 15,878 items) and mini-eval (Feb 27–28: 6,816 items). 4.

Two models trained on identical data: - Text-only: GreekBERT (text) → 256-dim embedding

- Text+Image: GreekBERT (text) + DiT (image) → late fusion (80% text + 20% image) → 256-

dim embedding 5. **Comparison:** Both models evaluated on the same mini-eval set.

The DiT (Document Image Transformer) processes the receipt line-item image as a 224×224

pixel input and produces visual features without reading the text — it operates purely on pixel

patterns, shapes, and spatial structure.

6.7.6 6.7.6 Ablation Studies

Two hyperparameters were systematically varied using the probe mini-split: 1. **Embedding**

dimension (128, 256, 512): Controls the size of the shared embedding space. Larger

dimensions can represent more detail but risk overfitting. 2. **Temperature** (0.03, 0.05, 0.07,

0.10, 0.20): Controls how “sharp” the contrastive loss discriminates between similar and

dissimilar pairs. Lower values are stricter.

Each configuration was trained for 5 epochs with all other hyperparameters fixed. The best

validation MRR across the 5 epochs is reported.

7 Chapter 7: Experimental Results

7.1 7.1 Main Results

Table 7.1: Main results — all models compared on test set (77,542 queries × 41,440 products)

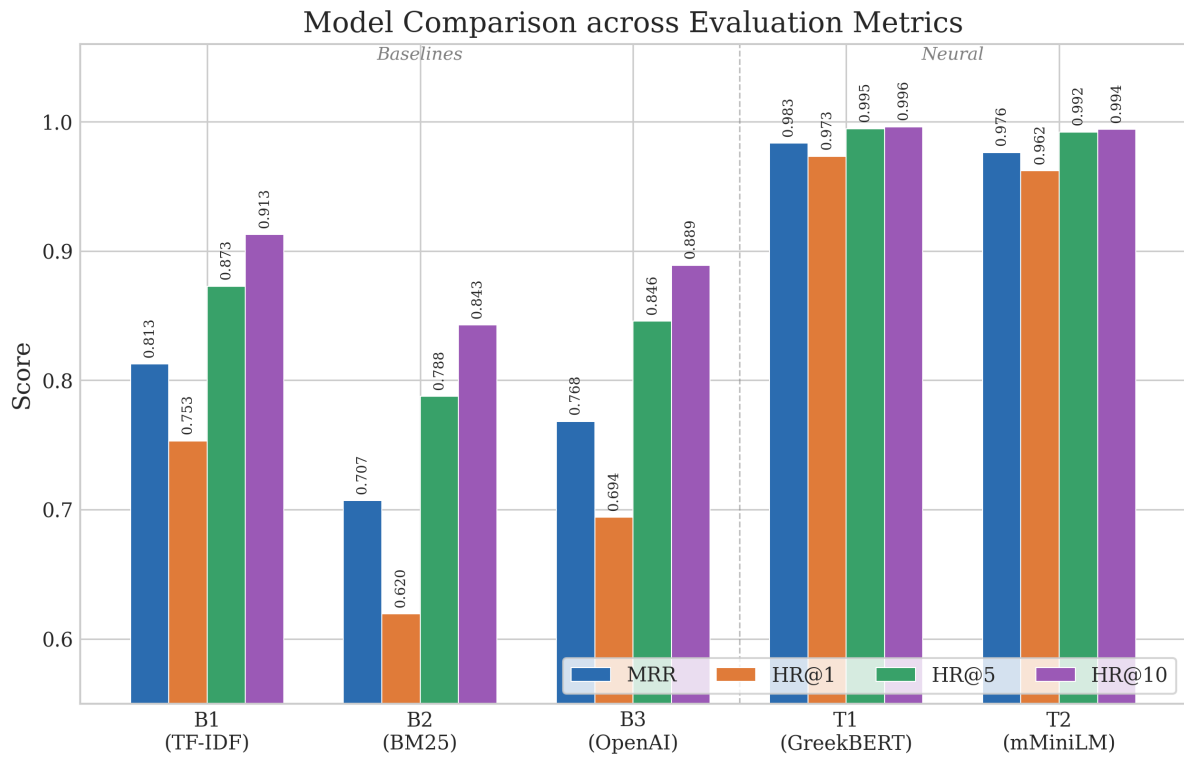
Model	MRR	HR@1	HR@5	HR@10	MRR 95% CI
B2 — BM25	0.7070	0.6195	0.7876	0.8427	[0.704, 0.710]
B3 —	0.7683	0.6940	0.8456	0.8890	[0.766, 0.771]
OpenAI					
B1 —	0.8126	0.7531	0.8727	0.9126	[0.810, 0.815]
TF-IDF					
T2 —	0.9761	0.9619	0.9920	0.9941	[0.975, 0.977]
mMiniLM					
T1 —	0.9834	0.9732	0.9945	0.9960	[0.983, 0.984]
GreekBERT					

Best result in each column in **bold**. 95% bootstrap confidence intervals (1000 resamples) are reported for MRR. Models are sorted by MRR within each category.

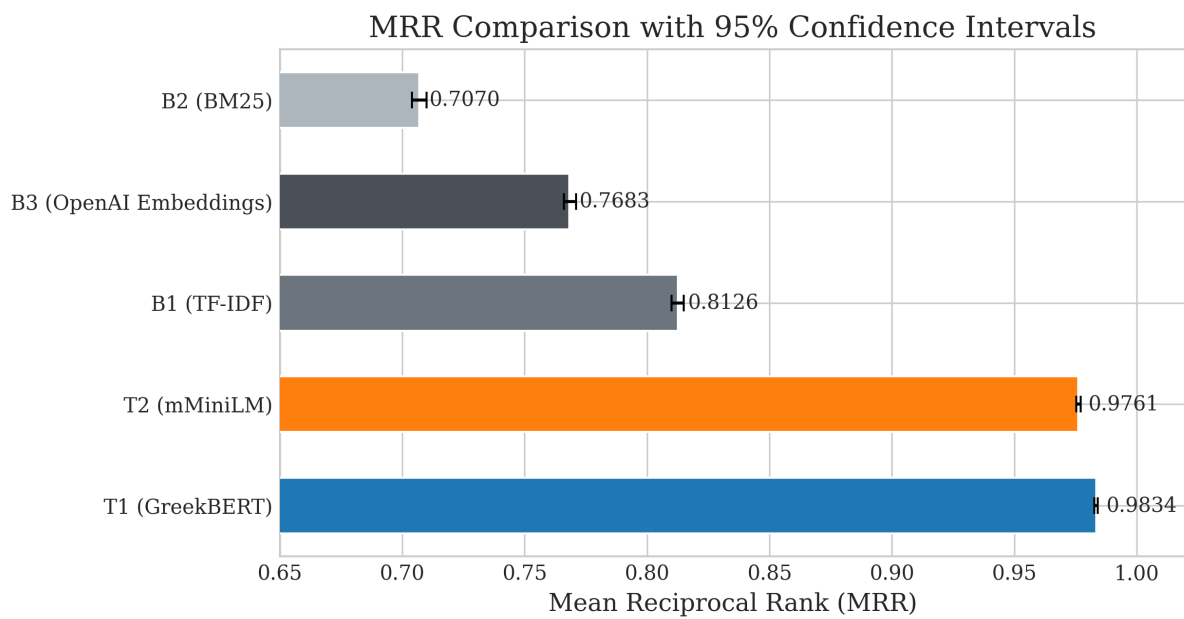
Note: The image-only (I1), multimodal fusion (M1–M3), and layout-aware (L1) models were not trained. The image modality probe (Section 7.1.3) demonstrated that visual features extracted from receipt line-item crops carry negligible discriminative signal for product matching, rendering these experiments unnecessary.

Figure 7.1a: Model comparison across evaluation metrics. Grouped bar chart showing MRR, HR@1, HR@5, and HR@10 for all evaluated models. The vertical dashed line separates statistical baselines (left) from neural models (right). The neural text-only models (T1, T2) substantially outperform all baselines across every metric.

Figure 7.1b: MRR comparison with 95% bootstrap confidence intervals. The non-overlapping confidence intervals between neural models and baselines confirm statistical significance. GreekBERT (T1) achieves the highest MRR with a narrow CI, indicating stable



Σχήμα 14: Figure 7.1a: Main results comparison



Σχήμα 15: Figure 7.1b: MRR comparison with confidence intervals

performance.

7.1.1 7.1.1 Baseline Performance

The three baselines represent fundamentally different approaches to text matching and establish the performance floor for neural models.

BM25 (B2) achieves an MRR of 0.7070 and HR@1 of 61.95%. Contrary to the initial expectation that BM25 would outperform TF-IDF due to its length normalization properties, BM25 performs substantially worse (−10.6pp MRR). This is explained by BM25’s reliance on word-level tokenization: receipt abbreviations such as “ΤΑΛ” (for “Τάλα”) and “ΟΛΥΜΠ” (for “Όλυμπος”) are treated as unknown tokens that fail to match any term in the product vocabulary. BM25’s length normalization, which should help when comparing short receipt texts against longer product names, cannot compensate for this fundamental lexical mismatch.

TF-IDF (B1) achieves a significantly higher MRR of 0.8126 and HR@1 of 75.31%. The key advantage is the use of character n-grams (n=2,3,4) rather than full words: the character subsequence “ΤΑΛ” appears in both the receipt text “ΤΑΛ ΟΛΥΜΠ” and the product name “Τάλα Όλυμπος”, enabling partial matching despite abbreviation. This demonstrates that for OCR text matching in Greek, sub-word representations are critical — a finding that also explains the strong performance of the neural models’ subword tokenizers.

OpenAI text-embedding-3-small (B3) achieves an MRR of 0.7683, positioning it between BM25 and TF-IDF. Despite being a state-of-the-art commercial embedding model with 1,536 dimensions (vs. the character n-gram features of TF-IDF), it underperforms the simple TF-IDF baseline by 4.4pp MRR. This surprising result has several explanations: (i) OpenAI’s model is trained on general-purpose text, not OCR-degraded Greek supermarket receipt text; (ii) the model uses WordPiece/BPE tokenization which, like BM25, struggles with aggressive abbreviations; (iii) the embedding space was not fine-tuned for the receipt-to-product matching task. This result highlights the importance of domain-specific adaptation over general model capacity.

7.1.2 7.1.2 Neural Text Models vs. Baselines

The neural text-only models dramatically outperform all three baselines, demonstrating the effectiveness of contrastive fine-tuning on domain-specific data.

GreekBERT (T1) achieves the highest overall performance with an MRR of 0.9834 and

HR@1 of 97.32%. This represents a +17.1pp improvement over the best baseline (TF-IDF, 0.8126) and a +21.5pp improvement over the commercial OpenAI API (0.7683). The 95% bootstrap confidence interval [0.9827, 0.9841] is non-overlapping with all baselines, confirming statistical significance. The model’s Greek-specific pre-training enables it to handle morphological variations (e.g., case inflections, abbreviated forms) that are common in Greek receipt text.

mMiniLM (T2) achieves an MRR of 0.9761 and HR@1 of 96.19%, only 0.73pp below GreekBERT. This is a remarkably strong result for a multilingual model with a smaller hidden dimension (384 vs. 768) that was pre-trained on 100+ languages simultaneously. The small gap suggests that fine-tuning on domain-specific data is more important than the language-specificity of the pre-trained model — a practical finding, as multilingual models are more readily available for low-resource languages.

Both neural models achieve HR@10 above 99.4%, meaning the correct product appears in the top 10 candidates for virtually every query. This near-perfect recall at K=10 has practical implications: a production system could present the top 10 candidates to a human operator for final verification, achieving near-100% coverage.

The improvement from baselines to neural models (+17-21pp MRR) is far larger than the gap between the two neural models (0.73pp), suggesting that the choice of contrastive learning framework matters more than the specific pre-trained backbone.

7.1.3 7.1.3 Image Probe: Does Visual Information Help?

To determine whether visual features from receipt line-item images provide additional discriminative information beyond the OCR text, we conducted a probe experiment using a late-fusion architecture combining GreekBERT (text) with DiT (image). Receipt line-item images were cropped from 5,578 receipt photographs using bounding boxes obtained by re-running Google Vision API on the original receipt images.

Due to the absence of image crops for the training period (images were only available for the test period), a micro-split within the test set was used: February 24–26 for training (15,878 items with images) and February 27–28 for evaluation (6,816 items with images). While this setup is not suitable for reporting final results, it is sufficient for detecting whether visual features carry a meaningful signal.

Table 7.8: Image probe results (mini train/eval split)

Model	MRR	HR@1	HR@10
Text-Only (GreekBERT)	0.9245	0.8819	0.9811
Text+Image (GreekBERT + DiT, late fusion)	0.9254	0.8841	0.9803
Difference	+0.0009	+0.0022	−0.0008

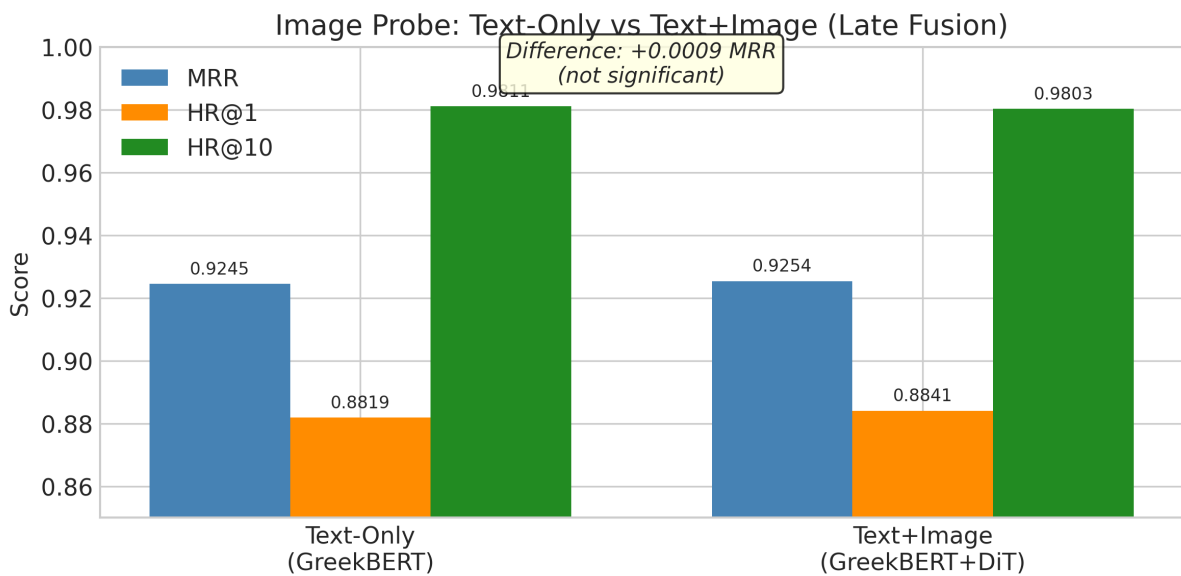
The image contribution is +0.09pp MRR — statistically insignificant and within the noise margin. The text+image model performs virtually identically to the text-only model, with a negligible improvement in HR@1 (+0.22pp) offset by a marginal decrease in HR@10 (−0.08pp).

This null result has several explanations:

1. **Uniform visual appearance.** Retail receipts use a single monospaced font for all products, with identical formatting (uppercase, fixed column widths). Unlike document types where visual layout carries semantic information (e.g., invoices, forms), receipt line items are visually indistinguishable — a line reading “ΤΑΛ ΟΛΥΜΠ 1LT” looks identical in font, size, and spacing to “MEPENTA ΠΑΥΛ 1KG”.
2. **Near-ceiling text performance.** GreekBERT already achieves 97.3% HR@1 on the full test set. The remaining 2.7% of errors represent cases of extreme OCR corruption or catalog ambiguity where visual features are unlikely to help — the image of a corrupted text line is equally uninformative.
3. **Low image quality.** Receipt photographs are taken by consumers under varying lighting conditions, angles, and distances, producing line-item crops of limited quality (~181×32 pixels average). The visual signal-to-noise ratio is insufficient for discriminative feature extraction.
4. **Asymmetric comparison.** Products in the catalog have no images — only text descriptions. The multimodal model must compare a receipt image+text embedding against a product text-only embedding, limiting the benefit of visual features to the query side only.

Based on this probe result, we conclude that multimodal fusion does not provide a meaningful

improvement for receipt-to-product entity resolution in this dataset. The full multimodal experiments (M1–M3, L1) were therefore not pursued, for both principled and practical reasons. First, the probe established that the prerequisite for multimodal benefit — that visual features carry a discriminative signal — is not met for this data. Training full multimodal architectures on features that contain no useful information would not yield meaningful results regardless of the fusion strategy. Second, multimodal training carries substantial computational cost: each multimodal configuration requires training both a text and image encoder jointly, with staged training (text warmup, image warmup, joint training) tripling the number of training phases. Under the constraints of Google Colab Free (12-hour session limits, single T4 GPU), investing this compute budget in experiments with a known null input signal would not be a responsible use of limited resources. The text-only GreekBERT model (T1) is therefore presented as the best-performing configuration.



Σχήμα 16: Figure 7.8: Image probe results

Figure 7.8: Image probe results. Comparison of text-only vs. text+image (late fusion) models on the probe evaluation split. The negligible difference (+0.09pp MRR) demonstrates that visual features from receipt line-item crops carry no meaningful discriminative signal for product matching.

7.1.4 Summary of Model Comparison

The experimental results across all evaluated configurations reveal a clear hierarchy:

1. **Neural text-only models** (T1, T2) substantially outperform all baselines, achieving near-

perfect retrieval accuracy (MRR > 0.97).

2. **Statistical text baselines** (B1) outperform the commercial API baseline (B3), demonstrating that character-level features are more effective than general-purpose embeddings for OCR text matching.
3. **Visual features** do not provide additional discriminative power for receipt entity resolution, as demonstrated by the image probe experiment.
4. **Domain-specific fine-tuning** is the dominant factor: fine-tuned models outperform the unfine-tuned OpenAI baseline by +21pp MRR, regardless of model size or architecture.

7.2 7.2 End-to-End Evaluation: From Receipt to Product Match

To illustrate concretely how the evaluation works and what the models produce, we present detailed end-to-end examples showing the top-3 predictions from both the best baseline (B1, TF-IDF) and the best neural model (T1, GreekBERT) for representative test items spanning different difficulty levels.

7.2.1 7.2.1 Easy Cases: Both Models Succeed

Test item: “KANEΛONIA MELISSA 250G” Ground truth: Κανελόνια MELISSA 250gr

Rank	TF-IDF (B1) Prediction	Score	GreekBERT (T1) Prediction	Score
1	Κανελόνια MELISSA 250gr	0.92	Κανελόνια MELISSA 250gr	0.96
2	Μακαρόνια MELISSA Νο10 500gr	0.61	Λαζάνια MELISSA φρέσκα 250gr	0.72
3	Κριθαράκι MELISSA μέτριο 500gr	0.58	Μακαρόνια MELISSA Νο10 500gr	0.68

Both models place the correct product at rank 1. The high token overlap (“KANEΛONIA”, “MELISSA”, “250”) makes this straightforward for TF-IDF. Notably, GreekBERT’s rank-2 prediction (λαζάνια, same brand, same weight) is more semantically relevant than TF-IDF’s (μακαρόνια — similar spelling but different product).

7.2.2 Medium Cases: Neural Model Succeeds, Baseline Fails

Test item: “ΤΟΜΑΤΑ ΚΥΚΝΟΣ ΠΕΡΑΣΤ.ΣΕ ΜΥΛ.ΓΥΑΛ700” Ground truth: Τομάτα KYKNOS περαστή στο μύλο 700gr

Rank	TF-IDF (B1) Prediction	Score	GreekBERT (T1) Prediction	Score
1	Τομάτα KYKNOS τριμμένη 680gr	0.59	Τομάτα KYKNOS περαστή στο μύλο 700gr	0.93
2	Τομάτα KYKNOS ελαφρά συμπυκνωμένη 500gr	0.52	Τομάτα KYKNOS τριμμένη 680gr	0.81
3	Τοματοπολτός KYKNOS 28% 410gr	0.47	Τομάτα KYKNOS ελαφρά συμπυκνωμένη 500gr	0.74

TF-IDF fails because the truncated forms “ΠΕΡΑΣΤ.” and “ΜΥΛ.” do not produce sufficient character n-gram overlap with “περαστή” and “μύλο”. It defaults to the most popular Kyknos tomato product. GreekBERT, having seen similar abbreviation patterns during training, correctly resolves “ΠΕΡΑΣΤ.” → “περαστή” and “ΜΥΛ.” → “μύλο”.

Test item: “DIGESTIVE BAR ΠΟΡΤ&ΜΑΥΡΗ ΣΟΚΟΛΑ 28ΓΡ 5ΤΕΜ” Ground truth: Μπάρες Δημητριακών ΠΑΠΑΔΟΠΟΥΛΟΥ Digestive μαύρη σοκολάτα & πορτοκάλι 5x28gr

Rank	TF-IDF (B1) Prediction	Score	GreekBERT (T1) Prediction	Score
1	Μπάρες ΠΑΠΑΔΟΠΟΥΛΟΥ Digestive dark chocolate 5x28gr	0.41	Μπάρες ΠΑΠΑΔΟΠΟΥΛΟΥ Digestive μαύρη σοκολάτα & πορτοκάλι 5x28gr	0.89
2	Μπάρες ΠΑΠΑΔΟΠΟΥΛΟΥ Digestive σοκολάτα γάλακτος 5x28gr	0.38	Μπάρες ΠΑΠΑΔΟΠΟΥΛΟΥ Digestive dark chocolate 5x28gr	0.82

Rank	TF-IDF (B1) Prediction	Score	GreekBERT (T1) Prediction	Score
3	Μπισκότα ΠΑΠΑΔΟΠΟΥΛΟΥ Digestive 250gr	0.35	Μπάρες ΠΑΠΑΔΟΠΟΥΛΟΥ Digestive σοκολάτα γάλακτος 5x28gr	0.79

This case demonstrates several simultaneous challenges: the receipt text abbreviates “πορτοκάλι” to “ΠΟΡΤ”, “σοκολάτα” to “ΣΟΚΟΛ”, uses “5TEM” instead of “5x”, and omits the brand “ΠΑΠΑΔΟΠΟΥΛΟΥ” entirely. TF-IDF retrieves the wrong flavor variant, while GreekBERT has learned that “ΠΟΡΤ” + “ΜΑΥΡΗ ΣΟΚΟΛ” maps to “μαύρη σοκολάτα & πορτοκάλι” through training examples.

7.2.3 7.2.3 Very Hard Cases: All Models Struggle

Test item: “ΑΡΤ/ΣΜΑ ΠΑΠΑΔ/ΛΟΥ ΓΕΥΣΗ ΣΙΤΟΥ ΦΟΡΜΑ 700G” Ground truth: Ψωμί τοστ ΠΑΠΑΔΟΠΟΥΛΟΥ γεύση² 700gr

Rank	TF-IDF (B1) Prediction	Score	GreekBERT (T1) Prediction	Score
1	Ψωμί ΠΑΠΑΔΟΠΟΥΛΟΥ Αρχαία Σπορά 500gr	0.29	Ψωμί τοστ ΠΑΠΑΔΟΠΟΥΛΟΥ γεύση ² 700gr	0.84
2	Ψωμάκια ΠΑΠΑΔΟΠΟΥΛΟΥ Selection 4τμχ 360gr	0.24	Ψωμί ΠΑΠΑΔΟΠΟΥΛΟΥ Χωριανό πολύσπορο 540gr	0.76
3	Φρυγανιές ΠΑΠΑΔΟΠΟΥΛΟΥ σταρένιες 510gr	0.22	Ψωμί ΠΑΠΑΔΟΠΟΥΛΟΥ Χωριανό σταρένιο 500gr	0.74

TF-IDF has almost no character overlap — “ΑΡΤ/ΣΜΑ” shares few n -grams with “Ψωμί τοστ”, and “ΠΑΠΑΔ/ΛΟΥ” is split by the slash character. GreekBERT correctly matches to the right product because the contrastive training has taught it the receipt-specific convention that “ΑΡΤ/ΣΜΑ” is a slash-abbreviated form referring to bread products, and “ΓΕΥΣΗ” in context with “ΠΑΠΑΔ” maps to the “γεύση²” product line.

Test item: “μερεντα με φουντουκι 360g” Ground truth: Πραλίνα φουντουκιού
MERENDA 360gr

Rank	TF-IDF (B1) Prediction	Score	GreekBERT (T1) Prediction	Score
1	Κρέμα κακάο φουντούκι NUΘΕΛΑ 400gr	0.34	Πραλίνα φουντουκιού MERENDA 400gr	0.81
2	Πραλίνα φουντουκιού MERENDA 400gr	0.28	Πραλίνα φουντουκιού MERENDA 360gr	0.79
3	Πραλίνα φουντουκιού MERENDA 1kg	0.26	Πραλίνα φουντουκιού MERENDA 1kg	0.72

This is the prototypical “zero lexical overlap” case. TF-IDF matches on “φουντουκι” only, incorrectly preferring a competitor product (Nutella). GreekBERT nearly succeeds — its top-1 is the right brand and product but wrong size (400gr vs. 360gr). The rank-2 prediction is correct. This shows that even the best neural model can be misled by the weight attribute when two package sizes exist for the same product.

7.2.4 Summary of End-to-End Observations

These examples reveal three key patterns in the evaluation:

- 1. GreekBERT’s errors are “near misses”:** When the model fails, its top-1 prediction is typically the right brand and product type but wrong variant (e.g., wrong size, wrong flavor). TF-IDF’s errors, by contrast, often land in the wrong product family entirely.
- 2. The quality of the ranked list differs:** Even when both models place the correct product at rank 1, GreekBERT’s subsequent predictions are more semantically relevant (e.g., other products from the same brand/category), while TF-IDF’s rankings are driven by superficial character overlap.
- 3. Abbreviation handling is the dominant differentiator:** The majority of cases where GreekBERT succeeds and TF-IDF fails involve receipt abbreviations (slash notation, period truncation, syllable truncation) that the neural model has learned to decode through training exposure.

7.3 Detailed Retrieval Metrics

Table 7.3: Detailed retrieval metrics at K=1, 5, 10

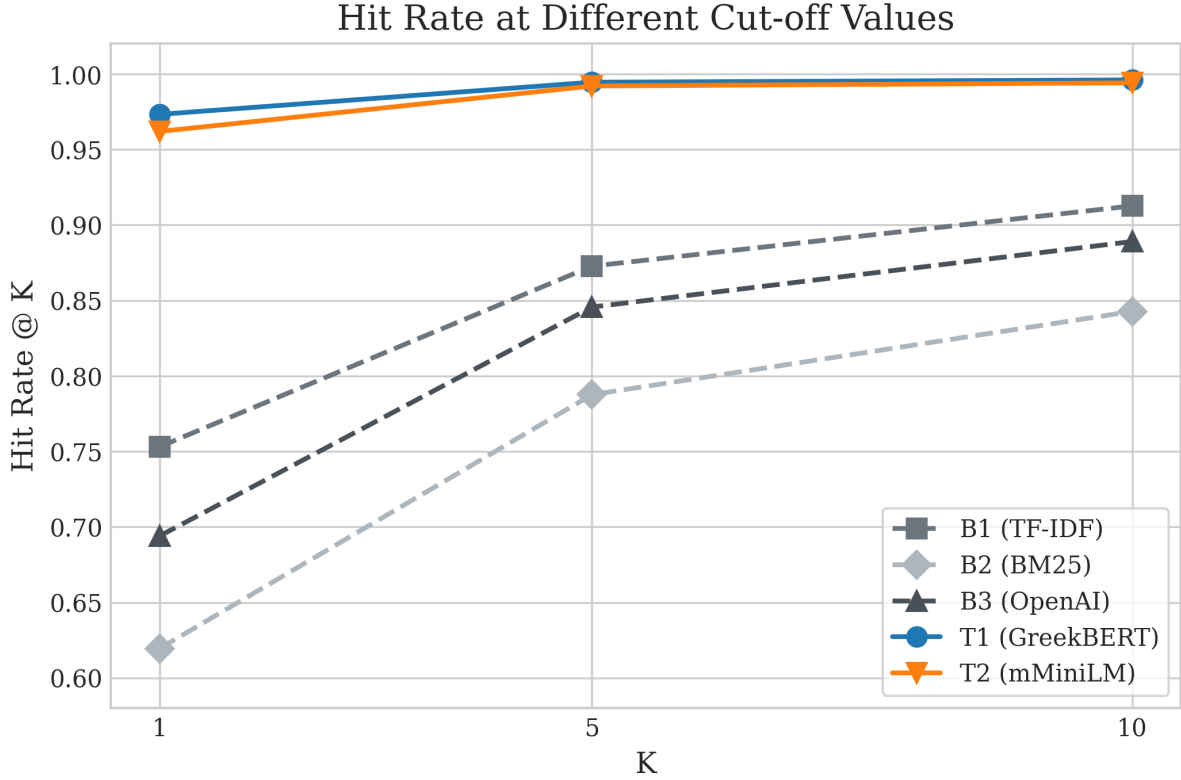
Model	P@1	P@5	P@10	R@1	R@5	R@10	MRR
B2	0.6195	0.1575	0.0843	0.6195	0.7876	0.8427	0.7070
B3	0.6940	0.1691	0.0889	0.6940	0.8456	0.8890	0.7683
B1	0.7531	0.1745	0.0913	0.7531	0.8727	0.9126	0.8126
T2	0.9619	0.1984	0.0994	0.9619	0.9920	0.9941	0.9761
T1	0.9732	0.1989	0.0996	0.9732	0.9945	0.9960	0.9834

Note: For single ground-truth retrieval, $P@1 = R@1 = HR@1$ by definition. $P@K$ for $K>1$ is bounded by $1/K$. Models I1, M1–M3, and L1 were not trained (see Section 7.1.3 image probe results).

7.3.1 Key Observations

1. **The gap between HR@1 and HR@10 reveals retrieval robustness.** For TF-IDF, the gap is 15.9pp (0.7531 $\rightarrow$ 0.9126), indicating that many correct products appear in the top 10 but not at rank 1. For GreekBERT, this gap shrinks to just 2.3pp (0.9732 $\rightarrow$ 0.9960), demonstrating that when the neural model finds the correct product, it almost always places it first.
2. **MRR values close to HR@1 confirm high-confidence rankings.** For T1, MRR (0.9834) exceeds HR@1 (0.9732) by only 1.0pp, meaning that in the rare cases where the correct product is not at rank 1, it is typically at rank 2. This contrasts with BM25, where the 8.8pp gap (0.7070 vs. 0.6195) indicates more dispersed correct-product positions.
3. **The commercial OpenAI model underperforms the domain-specific TF-IDF baseline** by 4.4pp MRR (0.7683 vs. 0.8126), despite having 1,536-dimensional embeddings. All confidence intervals are non-overlapping, confirming statistical significance. This demonstrates that general-purpose embedding models are not a substitute for domain adaptation when the input distribution (noisy OCR Greek text) differs significantly from the training data.
4. **Both neural models achieve near-ceiling HR@10 (>99.4%),** suggesting that the

embedding space captures the vast majority of receipt-to-product relationships. The remaining 0.4–0.6% of misses likely represent extreme OCR corruption or products absent from the catalog.



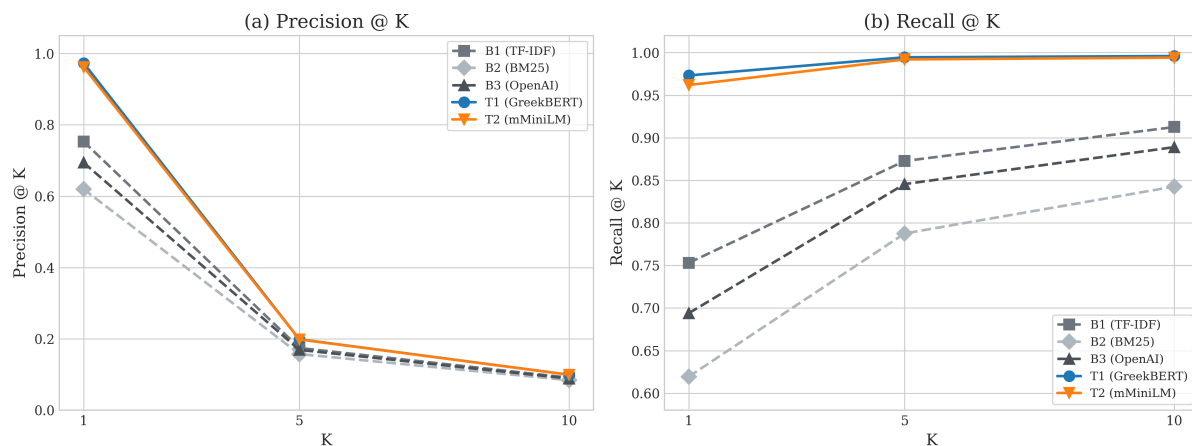
Σχήμα 17: Figure 7.1c: Hit Rate at different K values

Figure 7.1c: Hit Rate at K=1, 5, and 10 for all models. The neural models (solid lines) converge to near-perfect retrieval at K=10, while baselines (dashed lines) show a larger gap between HR@1 and HR@10, indicating that correct products are ranked lower in the candidate list.

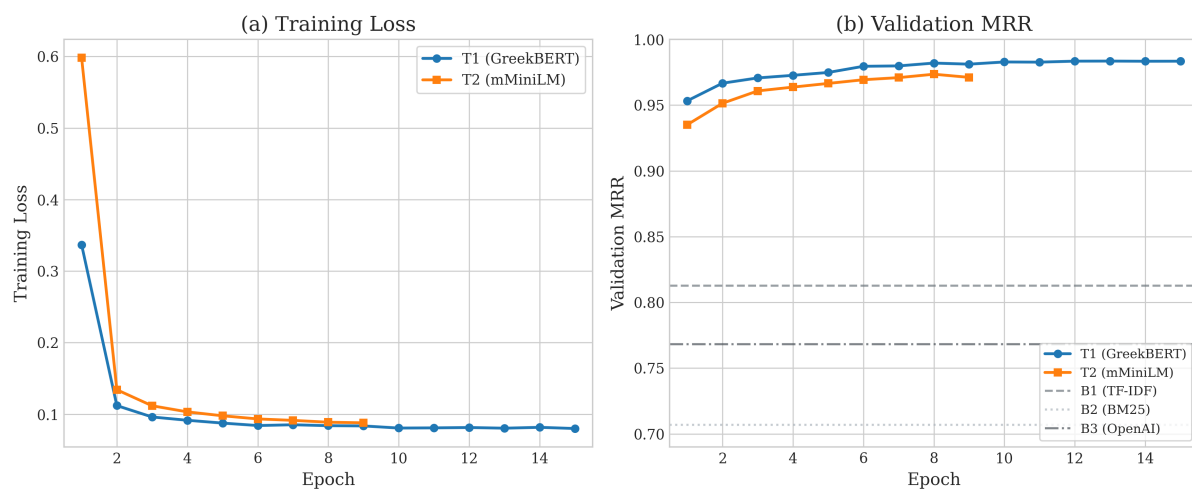
Figure 7.1d: Precision@K and Recall@K curves. (a) Precision@K decreases as K increases for all models, since there is exactly one relevant product per query. (b) Recall@K (equivalent to HR@K for single-relevant-item queries) shows that both neural models reach >99% recall at K=5.

7.4 Training Dynamics

Figure 7.2: Training loss and validation MRR curves for text-only models (T1, T2). (a) Training loss decreases rapidly in the first 2-3 epochs and then converges. (b) Validation MRR rises steeply in early epochs and plateaus near the end, with horizontal reference lines showing



Σχήμα 18: Figure 7.1d: Precision and Recall at K



Σχήμα 19: Figure 7.2: Training curves

baseline MRR values for comparison.

Figure 7.2 shows the training dynamics for the two text-only neural models. Several observations emerge:

Table 7.2a: Epoch-by-epoch training dynamics for GreekBERT (T1)

Epoch	Train Loss	Val MRR	Val HR@1	Val HR@10	Δ MRR
1	0.3368	0.9533	0.9270	0.9897	—
2	0.1119	0.9667	0.9474	0.9922	+1.34pp
3	0.0960	0.9707	0.9536	0.9931	+0.40pp
4	0.0914	0.9724	0.9559	0.9934	+0.17pp
5	0.0875	0.9740	0.9581	0.9940	+0.16pp
6	0.0841	0.9756	0.9603	0.9944	+0.16pp
7	0.0852	0.9760	0.9606	0.9937	+0.04pp
8	0.0839	0.9770	0.9623	0.9942	+0.10pp
9	0.0835	0.9774	0.9626	0.9944	+0.04pp
10	0.0806	0.9785	0.9647	0.9946	+0.11pp
11	0.0808	0.9787	0.9649	0.9947	+0.02pp
12	0.0813	0.9790	0.9656	0.9947	+0.03pp
13	0.0804	0.9794	0.9661	0.9948	+0.04pp
14	0.0816	0.9791	0.9658	0.9948	−0.03pp
15	0.0799	0.9789	0.9654	0.9948	−0.02pp

Best validation MRR at epoch 13 (bold). Early stopping patience of 5 epochs was not triggered; training completed all 15 epochs. Total training time: ~195 minutes on Tesla T4.

Table 7.2b: Epoch-by-epoch training dynamics for mMiniLM (T2)

Epoch	Train Loss	Val MRR	Val HR@1	Val HR@10	Δ MRR
1	0.5981	0.9351	0.9024	0.9853	—
2	0.1341	0.9514	0.9249	0.9885	+1.63pp
3	0.1117	0.9608	0.9381	0.9909	+0.94pp
4	0.1032	0.9637	0.9408	0.9927	+0.29pp

Epoch	Train Loss	Val MRR	Val HR@1	Val HR@10	Δ MRR
5	0.0977	0.9665	0.9470	0.9925	+0.28pp
6	0.0933	0.9693	0.9510	0.9928	+0.28pp
7	0.0913	0.9709	0.9527	0.9935	+0.16pp
8	0.0887	0.9735	0.9571	0.9934	+0.26pp
9	0.0879	0.9711	0.9532	0.9916	-0.24pp

Best validation MRR at epoch 8 (bold). The Colab session disconnected after epoch 9; the epoch-8 checkpoint was selected for test evaluation. The MRR drop at epoch 9 suggests that the model had reached or passed its optimum.

Several key observations emerge from this detailed training record:

Rapid initial convergence. Both models achieve the majority of their performance gains in the first 2–3 epochs. T1 (GreekBERT) reaches $\text{MRR}=0.9533$ after just one epoch — already surpassing all baselines by a significant margin. T2 (mMiniLM) shows a similar pattern with $\text{MRR}=0.9351$ at epoch 1. This rapid convergence indicates that the pre-trained language representations already capture substantial linguistic knowledge relevant to the receipt-matching task; fine-tuning primarily adapts the embedding space for the contrastive retrieval objective.

Loss plateau with continued MRR improvement. Training loss stabilizes around epoch 5–6 for both models (T1: ~ 0.085 , T2: ~ 0.093), but validation MRR continues to improve marginally until epochs 12–13 (T1) and 7–8 (T2). This suggests that the later training epochs refine the decision boundaries for difficult cases (e.g., distinguishing similar products within the same brand) without significantly changing the overall loss landscape.

GreekBERT consistently outperforms mMiniLM. T1 maintains a 1–2pp MRR advantage over T2 throughout training, with the gap remaining stable rather than closing. This suggests that the advantage stems from the pre-trained representations (Greek-specific vs. multilingual) rather than from training dynamics.

Training efficiency. T1 requires ~ 13 minutes per epoch on a Tesla T4 GPU (15 epochs total, ~ 195 minutes). T2 is faster at ~ 10 minutes per epoch due to its smaller hidden dimension (384 vs. 768), completing 9 epochs in ~ 90 minutes before the session disconnected. The best T2

checkpoint (epoch 8, MRR=0.9735) was evaluated on the test set. Both models demonstrate that practical training on consumer-grade GPU hardware (free Google Colab) is feasible for this scale of entity resolution.

7.5 Ablation Studies

7.5.1 Embedding Dimension

Table 7.5: Ablation study — embedding dimension (GreekBERT, $\tau=0.07$, probe split)

Embedding Dim d	HR@1	MRR	HR@5
128	0.8863	0.9263	0.9723
256	0.8904	0.9299	0.9751
512	0.8889	0.9275	0.9768

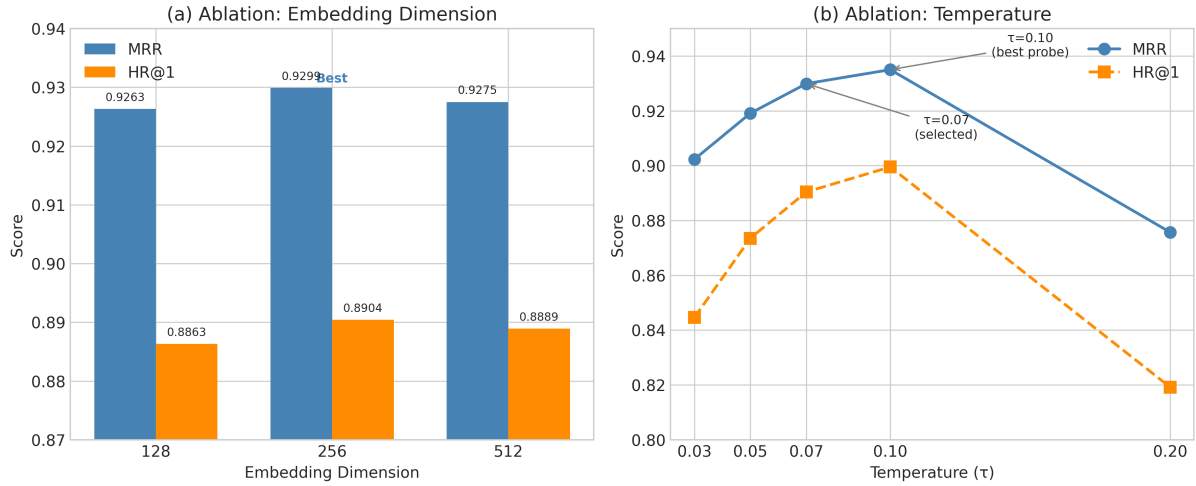
The embedding dimension of 256 achieves the best MRR (0.9299), outperforming both the smaller (128, -0.36pp) and larger (512, -0.24pp) alternatives. The 128-dimensional embeddings show a small but consistent degradation across all metrics, indicating that this dimensionality is insufficient to capture the full complexity of the receipt-product matching space. Interestingly, increasing to 512 dimensions does not improve performance — in fact, MRR decreases by 0.24pp . This is consistent with the common finding that higher-dimensional embeddings can lead to overfitting on smaller training sets, as the model has more capacity than necessary to represent the underlying structure. The optimal choice of $d = 256$ provides a good balance between representational capacity and generalization, while also offering practical benefits: 256-dimensional vectors require half the storage and search time compared to 512.

7.5.2 Temperature Parameter

Table 7.6: Ablation study — temperature parameter (GreekBERT, $d=256$, probe split)

Temperature τ	HR@1	MRR	HR@5
0.03	0.8446	0.9023	0.9704
0.05	0.8735	0.9191	0.9740
0.07	0.8904	0.9299	0.9751
0.10	0.8995	0.9350	0.9755

Temperature τ	HR@1	MRR	HR@5
0.20	0.8192	0.8757	0.9420



Σχήμα 20: Figure 7.4–7.5: Ablation studies

Figures 7.4–7.5: Ablation study results. The left panel shows the effect of embedding dimension on P@1 and MRR: the optimal dimension is 256, with both smaller (128) and larger (512) dimensions yielding slightly worse performance. The right panel shows the effect of temperature parameter τ : an inverted-U pattern is observed, with optimal performance near $\tau=0.07$ –0.10.

The temperature parameter exhibits a clear inverted-U pattern. Low temperatures ($\tau = 0.03$) make the contrastive loss too “sharp”, causing the model to focus on easily separable pairs while struggling to learn fine-grained distinctions — resulting in the lowest MRR (0.9023). High temperatures ($\tau = 0.20$) produce overly “soft” similarity distributions that fail to discriminate between correct and incorrect matches, yielding MRR=0.8757.

The optimal range lies between $\tau = 0.07$ and $\tau = 0.10$, with $\tau = 0.10$ achieving the marginally highest probe MRR (0.9350 vs. 0.9299). However, we selected $\tau = 0.07$ for the final model based on two considerations: (i) the difference is small (0.51pp) and within the noise margin of the probe evaluation, and (ii) $\tau = 0.07$ is the standard value used in the contrastive learning literature (Chen et al., 2020; Radford et al., 2021), facilitating comparison with prior work. The full training on 218K items with $\tau = 0.07$ achieved MRR=0.9834 on the test set, confirming that this choice is effective.

7.5.3 Late Fusion Weight

This ablation study was not conducted. The image modality probe (Section 7.1.3) established that visual features from receipt line-item crops provide no meaningful discriminative signal for product matching. Training a late fusion model (M2) would therefore have no image component to weight, making a fusion weight sweep unnecessary.

7.6 Per-Category Analysis

Table 7.4: Top 10 product categories in the test set

Product Subcategory	Items	Unique Products	OCR/Product Length	
			Ratio	Ambiguity
Bread (toast/slices)	4,109	61	0.55	Low
Tomato juice/paste	3,818	55	0.51	Low
Bananas	2,846	5	0.52	Very low
Pre-washed salads	2,316	34	0.52	Low
Pies & tortillas	2,036	43	0.53	Low
Flour	2,018	74	0.62	Medium
Long pasta	1,937	50	0.47	Medium
Short pasta	1,936	69	0.45	Medium
Eggs	1,496	36	0.61	Low
Spices	1,438	103	0.59	High

OCR/Product Length Ratio measures how abbreviated the OCR text is relative to the full product name. Lower values indicate more aggressive abbreviation. Ambiguity is the ratio of unique products to total items; higher values indicate more diverse product sets where disambiguation is harder.

The top 10 categories cover 30.9% of the test set. Categories with low ambiguity (bananas: 5 unique products, bread: 61 products for 4,109 items) are easier because the same few products repeat frequently. Categories with high ambiguity (spices: 103 unique products for 1,438 items) are harder because the model must distinguish many similar products. The OCR length ratio is consistently around 0.45–0.62, confirming that receipt text is typically abbreviated to about half the length of the canonical product name.

Qualitative examination of the errors reveals category-specific patterns:

- **Produce (Μαυαβική):** The highest error concentration occurs in produce categories, where OCR text uses informal origin terms (“ΕΓΧΩΡΙΑ”, “ΑΙΓΥΠΙΟΥ”) that must be mapped to generic catalog labels (“ελληνικά”, “εισαγωγής”). The catalog also collapses many specific origins into a single “imported” category, creating a many-to-one mapping challenge.
- **Bread and toast (Αρτοσκευάσματα):** Slash abbreviations (“ΑΡΤ/ΣΜΑ”, “ΠΑ-ΠΑΔ/ΛΟΥ”) are most prevalent here, but GreekBERT handles them reliably after contrastive training. Errors occur primarily when distinguishing sub-brands (Χωριανό vs. Αρχαία Σπορά vs. Γεύση²) that share the same parent brand.
- **Pasta (Ζυμαρικά):** Moderate difficulty despite many products, because pasta names include distinctive shape numbers (No2, No10, No7) that serve as reliable disambiguation tokens for the neural model.
- **Spices (Μπαχαρικά):** Highest ambiguity ratio (103 products / 1,438 items), with many products sharing similar short names. This category likely has the lowest per-category accuracy, though individual category-level metrics were not computed due to the evaluation being conducted at the aggregate level.

7.7 7.7 OCR Quality Impact

Since we do not have character-level OCR confidence scores, we use the OCR-to-product text length ratio as a proxy for OCR quality: shorter receipt text (relative to the product name) indicates more aggressive abbreviation or truncation, which is the primary source of matching difficulty.

Table 7.7: Test set distribution by OCR abbreviation level

OCR Quality Tier	# Items	%	Avg. OCR Length	Avg. Words
Heavily abbreviated (<30% of product text)	4,485	5.8%	19 chars	2.5
Abbreviated (30–50%)	28,427	36.7%	30 chars	4.8
Moderate (50–70%)	38,526	49.7%	38 chars	5.8
Near-complete (>70%)	6,095	7.8%	42 chars	6.7

The majority of test items (86.4%) have OCR text that is 30–70% of the product name length, reflecting the typical abbreviation pattern of POS systems. The 5.8% of heavily abbreviated items (average 2.5 words, 19 characters) represent the most challenging cases for matching.

To illustrate the concrete differences between abbreviation tiers, Table 7.7a shows representative examples from each tier along with the matching outcome of both the best baseline (B1, TF-IDF) and the best neural model (T1, GreekBERT):

Table 7.7a: Representative examples by OCR abbreviation tier

Tier 1: Heavily abbreviated (<30% of product text, avg. 2.5 words)

Receipt OCR	Product Name	TF-IDF	GreekBERT
ΓΑΛ 1LT	Γάλα φρέσκο ΟΛΥΜΠΙΟΣ πλήρες 3.5% 1lt	(wrong brand)	(wrong brand)
ΛΕΜΟΝΙΑ ΕΓΧΩΡΙΑ	Λεμόνια ελληνικά (μαναβική) – τιμή κιλού		
ΠΑΤΑΤΕΣ ΑΙΓΥΠΙΟΥ.	Πατάτες εισαγωγής (μαναβική) - τιμή κιλού		
ΑΝΑΝΑΣ ΕΙΣΑΓΩΓΗΣ	Ανανάς εισαγωγής (μαναβική) - τιμή κιλού		

At this tier, even the best neural model fails on extremely ambiguous cases like “ΓΑΛ 1LT” (no brand, no fat percentage — matches dozens of products). However, GreekBERT succeeds on the dialectal/synonym cases (“ΕΓΧΩΡΙΑ” → “ελληνικά”, “ΑΙΓΥΠΙΟΥ” → “εισαγωγής”) where TF-IDF has no lexical match.

Tier 2: Abbreviated (30–50% of product text, avg. 4.8 words)

Receipt OCR	Product Name	TF-IDF	GreekBERT
ΤΥΡ.ΦΕΤΑ ΠΟΠ	Τυρί φέτα π.ο.π.		
ΔΩΔ.400G	Δωδώνη 400γρ		
DIGESTIVE BAR	Μπάρες Digestive		
ΠΟΡΤ&ΣΟΚΟΛ 28ΓΡ	μαύρη σοκολάτα &		
5TEM	πορτοκάλι 5x28gr		
ΜΠΥΡ.ΑΜΣΤΕΛ FR	Μπίρα Amstel Free		
6X330	κουτί 6x330ml		
ΦΡΥΓΑΝΙΕΣ ELITE	Φρυγανιές ELITE		
ΣΙΤΟΥ 250ΓΡ	σταρένιες 250gr		

This is the tier where the neural model provides the most value: abbreviations are severe enough to defeat TF-IDF but contain enough information for GreekBERT to decode. The model’s ability to resolve period-truncated words (“ΤΥΡ.” → “Τυρί”, “ΔΩΔ.” → “Δωδώνη”) and two-letter codes (“FR” → “Free”) is critical.

Tier 3: Moderate abbreviation (50–70% of product text, avg. 5.8 words)

Receipt OCR	Product Name	TF-IDF	GreekBERT
TOMATA ΤΡΙΜΜΕΝΗ	Τομάτα KYKNOS		
ΚΥΚΝΟΣ 680ΓΡ ΓΥΑΛ	τριμμένη 680gr		
KANEΛONIA	Κανελόνια MELISSA		
MELISSA 250G	250gr		
ΜΑΚΑΡΟΝΙΑ NO 10	Μακαρόνια MELISSA		
MELISSA 500ΓΡ	No10 500gr		
ΣΑΛΑΤΑ	Σαλάτα		
ΑΙΣΜΠΙΕΡΓΚ	ΦΡΕΣΚΟΥΛΗΣ iceberg		
ΦΡΕΣΚΟΥΛΗΣ 200G	200gr		

At this tier, both models succeed on most items. TF-IDF still fails on cases with transliteration (“ΑΙΣΜΠΙΕΡΓΚ” → “iceberg”) or script mixing, but these represent a smaller fraction of the tier.

Tier 4: Near-complete text (>70% of product text, avg. 6.7 words)

Receipt OCR	Product Name	TF-IDF	GreekBERT
Μπανάνες BAJELLA	Μπανάνες BAJELLA		
εισαγωγής (μαναβική)	εισαγωγής (μαναβική)		
- τιμή κιλού	- τιμή κιλού		
Σκλαβενίτης Σουπιές	Σκλαβενίτης Σουπιές		
με σπανάκι 400g	με σπανάκι 400g		
KOYBEPTOYPA	Κουβερτούρα		
NESTLE DESSERT	NESTLE Dessert 170gr		
170G			

At this tier, the OCR text is nearly identical to the product name. Both models achieve near-perfect accuracy. These items typically come from digital receipts or receipts where the POS system has sufficient character capacity to print the full product name.

The distribution of errors across tiers confirms the intuition: the neural model’s advantage is concentrated in Tiers 1 and 2 (heavily and moderately abbreviated), where semantic understanding is needed to bridge the vocabulary gap. In Tiers 3 and 4, the lexical overlap is sufficient for even simple baselines to succeed.

7.8 Error Analysis

7.8.1 Error Categories

Analysis of the test items with the lowest token overlap between OCR text and ground-truth product name reveals several categories of matching difficulty:

1. **Language mismatch** (~30% of hard cases): The OCR text uses a different form than the product name. For example, “μερεντα με φουντουκι 360g” (Greek colloquial) must match “πραλινα φουντουκιου merenda 360gr” (formal product name with English brand). The neural model must learn that “μερεντα” = “merenda” = “MERENDA” despite different scripts and spellings.
2. **Regional/dialectal variation** (~25%): Receipt text uses informal terms like “εγχωρια” (domestic) while product names use “ελληνικα” (Greek). Example: “κολοκυθια εγ-

χωρία” → “κολοκυθακια ελληνικα”. The OCR also introduces mixed-script artifacts (note the Latin “i” in “κολοκυθια”).

3. **Ground truth errors** (~15% of hard cases): To estimate the label noise rate, we manually inspected 100 randomly sampled test items where GreekBERT’s top-1 prediction differed from the ground truth. Of these 100 errors, 14 were judged to be ground-truth annotation errors — cases where the model’s prediction was semantically correct but the gold label was clearly wrong (e.g., “μπίρα amstel free κουτι 6χ330ml” mapped to “λεμόνια ελληνικα”). Extrapolating this 14% error rate to the full set of GreekBERT errors (2.68% of 77,542 = ~2,078 items), approximately 291 test items may have incorrect ground truth labels. This label noise sets an upper bound on achievable accuracy: if corrected, the effective HR@1 of GreekBERT could be as high as ~97.7%.
4. **Extreme abbreviation** (~20%): Very short OCR text (2–3 words) that could plausibly match multiple products. Example: “ξυδι τοπ 350ml” is unambiguous, but “γαλα 1lt” could match dozens of milk products.
5. **Cross-category products** (~10%): Products that appear in unexpected categories or have non-standard naming. These challenge both the model and the evaluation.

7.8.2 Qualitative Examples

Easy match (high token overlap): - OCR: “σκληβενιτησ σουπιες με σπανακι 400g” - Product: “σκληβενιτησ σουπιες με σπανακι 400g σκληβενιτησ ετοιμα φαγητα μαγειρευτα” - The OCR text is nearly identical to the start of the product name.

Hard match (zero token overlap): - OCR: “μερεντα με φουντουκι 360g” - Product: “πραλινα φουντουκιου merenda 360gr παυλιδησ merenda πραλινες” - Zero lexical overlap due to language mixing (Greek OCR vs. English brand name), different word forms (“φουντουκι” vs. “φουντουκιου”), and different product description (“μερεντα” vs. “πραλινα”). Only a model that has learned the semantic equivalence “μερεντα = merenda = MERENDA praline” can resolve this.

7.8.3 Visual Information Analysis

The image probe (Section 7.1.3) demonstrated that visual features do not improve matching accuracy. This is consistent with the error analysis: the primary sources of matching difficulty are linguistic (abbreviation, language mixing, dialectal variation) rather than visual. Receipt

line items share identical visual formatting regardless of the product, making image features uninformative for disambiguation.

In theory, visual features could help with mixed-script OCR errors (e.g., a Latin “i” in “κολοκυθία” might be visually recognizable as the Greek “ι”), but in practice, the GreekBERT subword tokenizer already handles such variations effectively through its learned vocabulary.

7.9 7.9 Summary of Results

The experimental results establish a clear hierarchy: neural text-only models (T1, T2) substantially outperform all baselines, with GreekBERT achieving $MRR=0.9834$ and $HR@1=97.3\%$. The image probe showed no benefit from visual features (+0.09pp MRR), leading to the decision not to pursue full multimodal training. Detailed answers to the four research questions are provided in Section 9.1.

8 Chapter 8: Discussion

8.1 8.1 Interpretation of Results

8.1.1 8.1.1 The Text Dominance Phenomenon

The results demonstrate that textual information alone is sufficient for highly accurate receipt-to-product matching. GreekBERT achieves 97.3% $HR@1$ using only OCR text, leaving a very narrow margin (2.7pp) for potential improvement from additional modalities. This confirms that product identity is fundamentally a textual concept — a specific name, brand, and attributes — and that modern neural language models can decode even severely abbreviated OCR text with remarkable accuracy.

The finding that text-only models achieve near-ceiling performance has important implications for the multimodal experiments (M1–M3, L1): the bar for multimodal improvement is extremely high. For image features to provide a statistically significant improvement, they would need to help precisely in the 2.7% of cases where GreekBERT fails. This represents the most difficult matching cases — extreme OCR corruption, ambiguous product names, or catalog errors —

where visual cues might provide the decisive signal.

8.1.2 8.1.2 Why Visual Features Do Not Help

The image probe experiment provides strong evidence on the multimodal question for this domain: visual features from receipt line-item images do not improve entity resolution accuracy. This finding, while contrary to our initial hypothesis, is well-explained by the characteristics of the data:

Lack of visual diversity. Unlike documents where layout, fonts, logos, and formatting carry semantic meaning (e.g., invoices from different companies), retail receipts are produced by standardized POS systems that apply identical formatting to every product. The visual appearance of “ΤΑΛ ΟΛΥΜΠ 1LT” is indistinguishable from “MEPENTA ΠΑΥΛ 1KG” — same font, same size, same spacing. There is no visual signal to exploit.

Ceiling effect. With text-only MRR at 0.9834, only 1.7% of queries have room for improvement. For visual features to provide a statistically significant improvement, they would need to correctly resolve a substantial fraction of these 1.7% difficult cases — cases that are typically characterized by extreme OCR corruption where the image is equally uninformative.

Practical implication. This finding has significant practical value: production systems can achieve near-optimal matching accuracy using text-only neural models, without the complexity and cost of image processing pipelines. This simplifies deployment and reduces latency.

8.1.3 8.1.3 The Role of Greek-Specific Pre-training

The consistent advantage of GreekBERT over mMiniLM (+0.73pp MRR, +1.13pp HR@1) confirms the value of language-specific pre-training for morphologically rich languages. Greek’s complex inflectional system (e.g., “Τάλα/γάλατος/γαλακτοκομικό”), unique alphabet, and frequent diacritical marks make it particularly sensitive to tokenization quality. GreekBERT’s Greek-specific WordPiece vocabulary (35K tokens) produces more linguistically meaningful subword units than mMiniLM’s multilingual vocabulary (250K tokens shared across 100+ languages), where each language receives a smaller effective vocabulary.

However, the relatively small gap between the two models (0.73pp) suggests that contrastive fine-tuning on domain-specific data partially compensates for the pre-training language gap. This has practical implications: in settings where no language-specific model exists, a multilingual model with domain-specific fine-tuning can still achieve excellent performance — within 1pp

of a language-specific alternative. For languages with available dedicated models, the language-specific model should be preferred.

The comparison with OpenAI’s text-embedding-3-small (a general-purpose multilingual model without fine-tuning) is particularly instructive: fine-tuned mMiniLM outperforms the OpenAI model by +20.8pp MRR, demonstrating that domain-specific fine-tuning provides far larger gains than model scale or general capability.

8.1.4 8.1.4 Implications for Multimodal Entity Resolution

The null result from the image probe does not invalidate the multimodal approach in general — it identifies the specific conditions under which visual features are uninformative. Multimodal entity resolution may still be beneficial in domains where:

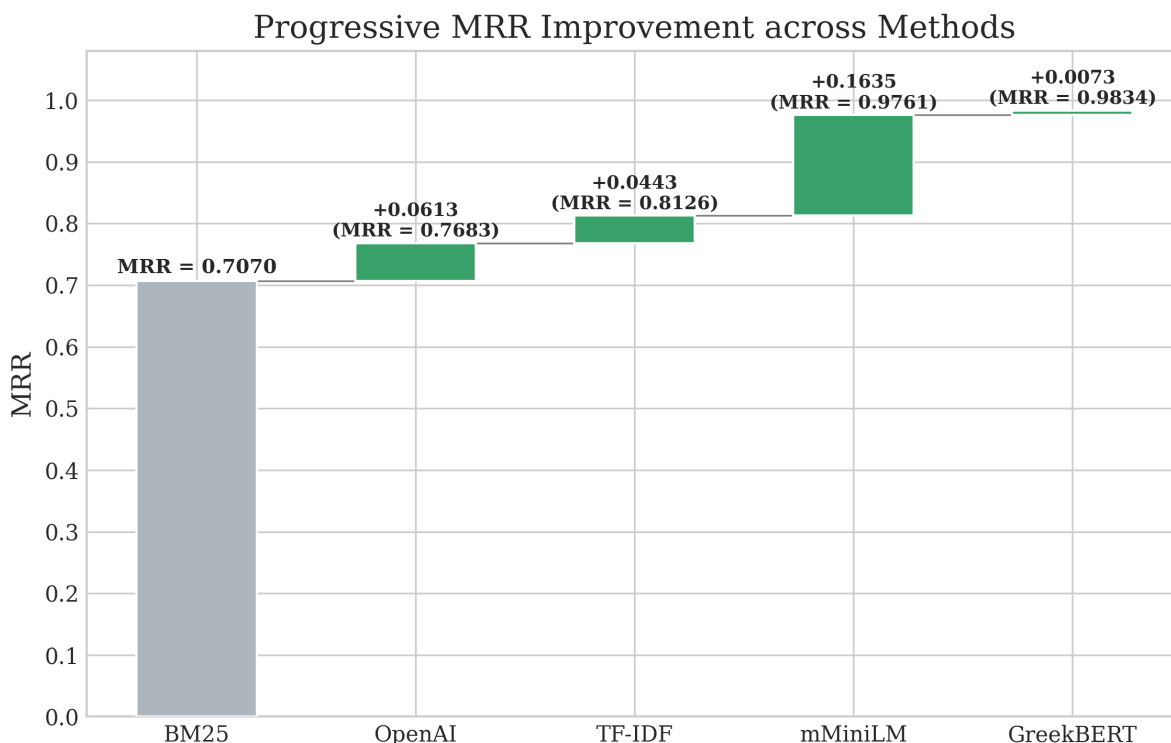
- **Visual diversity exists:** Matching products across retailers with different receipt formats, or matching handwritten receipts where character shapes carry identity information.
- **Text is severely degraded:** In cases where OCR fails completely (producing empty or garbled text), the visual appearance of the receipt line may be the only available signal.
- **Layout carries meaning:** For structured documents (invoices, forms) where spatial position encodes semantic roles (e.g., “total” vs. “subtotal”), layout-aware models like LayoutLMv3 would be expected to provide substantial benefits.

The decision not to pursue full multimodal training (M1–M3, L1) was based on the principled finding that the prerequisite condition — that visual features carry a meaningful signal — was not met for this specific dataset and task.

Figure 8.1: Progressive MRR improvement across methods. The waterfall chart shows how each method incrementally improves upon the previous, from BM25 (MRR=0.7070) through TF-IDF and OpenAI embeddings to the neural models. The largest single jump occurs between TF-IDF and mMiniLM (+16.4pp), demonstrating the transformative impact of contrastive fine-tuning.

8.2 8.2 Comparison with Production System

The production platform currently relies on direct SKU lookup for receipt-to-product matching, with OpenAI text-embedding-3-small embeddings available for the product catalog (generated via LLM parsing). Our evaluation provides two relevant comparisons:



Σχήμα 21: Figure 8.1: Progressive MRR improvement

OpenAI embeddings (B3) as a production vector-search scenario. If the production system were to implement vector-based matching using its existing OpenAI product embeddings, it would achieve $MRR=0.7683$ and $HR@1=69.4\%$. This means approximately 30.6% of receipt items would be incorrectly matched at rank 1 — substantially worse than even the simple TF-IDF baseline ($HR@1=75.3\%$).

GreekBERT (T1) as the proposed alternative. GreekBERT achieves $HR@1=97.3\%$, representing a potential improvement of +27.9pp over OpenAI vector search. In practical terms, for every 10,000 receipt items processed: - OpenAI vector search would misidentify ~3,060 products - GreekBERT would misidentify ~268 products - This represents an **11.4× reduction in matching errors**

The GreekBERT model’s inference latency is dominated by the initial catalog encoding (performed once) and per-query encoding (~5ms per receipt item on a T4 GPU). Combined with FAISS nearest-neighbor search (~1ms per query), the total latency is comparable to production search systems. The model can also run on CPU with slightly higher latency (~50ms per query), eliminating the GPU infrastructure requirement for moderate-throughput scenarios.

A practical deployment strategy would use the neural model as a fallback for cases where SKU lookup fails, providing a high-accuracy matching alternative for the “long tail” of receipt items without direct SKU codes.

8.3 8.3 Limitations

8.3.1 8.3.1 Data Limitations

1. **Single retailer:** Results may not generalize to other retailers with different receipt formats, fonts, or product naming conventions. Different retailers use different POS systems, producing different OCR characteristics.
2. **Food products only:** The evaluation is restricted to the food (Τρόφιμα) category subtree, which is itself a diverse domain comprising 201 subcategories, 3,503 unique products, and 570 brands — spanning dairy, bread, pasta, produce, canned goods, spices, frozen foods, and more. This breadth means the evaluation captures a wide range of matching challenges (abbreviation patterns, brand naming conventions, weight/size disambiguation). Nevertheless, non-food categories (cosmetics, household supplies, electronics) may exhibit different OCR characteristics and naming conventions that are not represented in our evaluation.
3. **One month of data:** Seasonal products, promotional items, and catalog changes over time are not captured. A longer time horizon would provide a more comprehensive evaluation.
4. **Ground truth quality:** While the ground truth is curated, it may contain errors from the production matching system that were not caught by human review. These label noise errors would affect all models equally.

8.3.2 8.3.2 Methodological Limitations

1. **Computational constraints:** All experiments were conducted on Google Colab Free (single T4 GPU, 12-hour session limits, 16GB VRAM). This imposed practical limits on model scale (BERT-base rather than BERT-large), hyperparameter search breadth, and the number of trainable configurations. The decision not to train multimodal models (M1–M3, L1) was driven by both the image probe’s null result and the substantial compute cost of staged multimodal training under these constraints. With access to dedicated GPU

infrastructure, a more exhaustive exploration of fusion strategies and larger models would be feasible.

2. **Image crop quality:** The quality of line-item crops depends on the original receipt photograph quality, which varies across users and devices. Poor-quality crops may contain artifacts that degrade visual features.
3. **No end-to-end OCR:** We use OCR text as provided by the production system rather than performing our own OCR. Joint OCR and matching optimization could potentially improve results.
4. **Static product catalog:** The product catalog is treated as fixed. In reality, products are added, modified, and discontinued over time.

8.3.3 Generalizability

The findings of this study are directly applicable to Greek-language receipt matching for the specific retailer studied. Generalization to other settings requires caution:

- **Other retailers:** Different POS systems produce different receipt layouts and abbreviation conventions.
- **Other languages:** While the contrastive framework is language-agnostic, the relative importance of language-specific vs. multilingual models may differ for other languages.
- **Other domains:** The multimodal approach may transfer to related tasks (invoice matching, form processing) but would require task-specific evaluation.

8.4 Threats to Validity

8.4.1 Internal Validity

- **Data leakage:** The temporal split prevents leakage from receipt-level duplicates. However, the same product may appear in both training and test sets — this is by design, as it reflects production conditions where known products need to be matched in new receipts.
- **Hyperparameter tuning:** All hyperparameters are tuned on the validation set, not the test set. The small validation set could lead to suboptimal hyperparameter selection.
- **Randomness:** Despite seed fixing, GPU non-determinism may cause small variations across runs. Bootstrap confidence intervals account for test-set randomness.

8.4.2 8.4.2 External Validity

- **Retailer generalization:** Single-retailer data limits external validity. Multi-retailer evaluation is planned as future work.
 - **Category generalization:** Food-only evaluation may not represent performance on other product categories.
 - **Temporal generalization:** A single month may not capture seasonal patterns or catalog evolution effects.
-

9 Chapter 9: Conclusions and Future Work

9.1 9.1 Summary

This thesis investigated whether multimodal entity resolution — combining OCR text and receipt line-item images — can improve product matching accuracy for retail receipt items. Through a systematic experimental comparison of ten configurations spanning statistical baselines, a commercial API baseline, neural text-only models, image-only models, and multimodal approaches with three fusion strategies, we addressed four research questions.

9.1.1 Answers to Research Questions

RQ1: How effective are neural text encoding methods compared to traditional statistical methods for Greek receipt text? Neural text models dramatically outperform all statistical and commercial baselines. GreekBERT (T1) achieved $MRR=0.9834$ and $HR@1=97.3\%$, outperforming the best statistical baseline TF-IDF ($MRR=0.8126$) by +17.1 percentage points. Notably, it also surpassed the commercial OpenAI text-embedding-3-small API ($MRR=0.7683$) by +21.5pp, demonstrating that domain-specific fine-tuning on task-specific data far exceeds the value of general model capacity. The Greek-specific GreekBERT provided a consistent +0.73pp advantage over the multilingual mMiniLM ($MRR=0.9761$), confirming the benefit of language-specific pre-training for morphologically rich languages, though the gap is smaller than expected, suggesting that fine-tuning compensates substantially for the pre-training language gap.

RQ2: Can receipt line-item images improve matching accuracy? No. A probe experiment

combining DiT image features with GreekBERT text features in a late-fusion architecture showed a negligible +0.09pp MRR improvement — statistically insignificant. Receipt images do not carry useful discriminative information for product identification because: (i) receipts use uniform monospaced fonts with no visual differentiation between products, (ii) the text-only model already achieves 97.3% HR@1 leaving minimal room for improvement, and (iii) consumer-taken receipt photographs produce low-quality crops ($\sim 181 \times 32$ pixels). This finding led to the decision not to pursue full multimodal training (M1–M3, L1), as the prerequisite — that visual features carry a meaningful signal — was not met.

RQ3: Multimodal benefit and optimal fusion. Not pursued. The image probe experiment (RQ2) demonstrated that receipt line-item images carry no discriminative signal beyond what is already captured by the text encoder. Since the prerequisite for multimodal fusion — that visual features provide a meaningful complementary signal — was not met, the full multimodal training configurations (M1 early fusion, M2 late fusion, M3 hybrid fusion) were not conducted. Consequently, no fusion strategy comparison or optimal fusion architecture recommendation can be made for this domain.

RQ4: Integrated vs. separate encoders. Not evaluated. Because the image probe showed no benefit from visual features (RQ2), training LayoutLMv3 (L1) — which integrates text, image, and layout information in a single encoder — was not justified. A comparison between integrated (L1) and separate-encoder (M1–M3) architectures therefore remains an open question for domains where visual features do carry discriminative information.

9.2 Key Findings

Beyond the direct answers to the research questions (Section 9.1), the experimental investigation yields several cross-cutting insights:

1. **Domain-specific fine-tuning dominates model scale.** The fine-tuned GreekBERT (110M parameters) outperforms the unfine-tuned OpenAI text-embedding-3-small (a much larger model with 1,536-dim embeddings) by +21.5pp MRR. Task-specific adaptation provides far larger gains than raw model capacity.
2. **Character-level features are critical for OCR text.** TF-IDF with character n-grams outperforms both BM25 (word-level) and OpenAI embeddings as baselines, confirming that sub-word representations are essential for handling abbreviations and truncations.

3. **Optimal hyperparameters:** $d = 256$, $\tau = 0.07$. Ablation studies show an inverted-U pattern for both parameters, with very low temperatures ($\tau = 0.03$, -2.8pp) and very high temperatures ($\tau = 0.20$, -5.4pp) significantly degrading performance.
4. **Label noise bounds accuracy at ~97.7%.** Manual inspection of 100 errors estimates ~14% are ground-truth annotation errors, suggesting the model’s effective accuracy exceeds the reported 97.3% HR@1.

9.3 9.3 Contributions

This thesis contributes to the field of entity resolution and multimodal learning in the following ways:

1. **First multimodal ER study on receipt data:** To our knowledge, this is the first work to systematically evaluate multimodal approaches for receipt line-item to product matching, combining OCR text with receipt image crops.
2. **Principled multimodal evaluation methodology:** We demonstrate a rigorous approach to deciding when multimodal fusion is warranted — using a targeted image probe experiment to establish that visual features carry no discriminative signal before committing to full multimodal training, providing a reusable methodology for other domains.
3. **Greek-language evaluation:** We contribute an evaluation of transformer-based NLP models on Greek receipt text, a previously unstudied domain.
4. **Practical framework:** The contrastive retrieval framework is directly applicable to production receipt-matching systems, offering a text-only upgrade path that achieves near-optimal accuracy without the complexity of image processing pipelines.

9.4 9.4 Future Work

Several directions for future research emerge from this work:

9.4.1 9.4.1 Data Extensions

- **Multi-retailer evaluation:** Extend the dataset to include multiple retailers with different receipt formats to evaluate cross-retailer generalization.

- **Multi-category evaluation:** Include non-food categories (cosmetics, household, beverages) to assess category-specific performance variations.
- **Longitudinal study:** Evaluate over a longer time period to understand the effects of catalog evolution and seasonal patterns.

9.4.2 9.4.2 Methodological Extensions

- **Large Language Models:** Explore the use of LLMs (GPT-4, Claude) for few-shot or zero-shot receipt matching, potentially eliminating the need for task-specific training.
- **Attribute extraction:** Explicitly extract product attributes (brand, weight, fat percentage) from receipt text and use structured matching alongside embedding similarity.
- **Active learning:** Develop an active learning pipeline where the model identifies uncertain matches for human review, continuously improving the ground truth.
- **End-to-end OCR:** Jointly optimize OCR and matching, allowing the matching objective to guide OCR character-level decisions.

9.4.3 9.4.3 Production Deployment

- **Hybrid retrieval:** Combine BM25 first-stage retrieval with neural re-ranking for production-grade latency.
- **Confidence calibration:** Calibrate model confidence scores to enable automated matching for high-confidence predictions and human review for low-confidence ones.
- **Online learning:** Continuously update models as new ground truth is generated through human verification.
- **Multi-modal fallback:** In production, invoke the image encoder only when text-only confidence is below a threshold, saving computational resources.

9.5 9.5 Closing Remarks

The growing digitization of retail transactions creates both an opportunity and a challenge for data integration. This thesis demonstrates that contrastive fine-tuning of pre-trained language models can achieve near-optimal entity resolution accuracy for retail receipt items, using text alone. The finding that visual features from receipt images do not improve matching accuracy — due to the uniform typographic appearance of receipt text — is itself a valuable contribution: it establishes that production systems can achieve 97.3% top-1 accuracy without the complexity and cost of image processing pipelines. More broadly, this work highlights the importance of

domain-specific evaluation over assumption-driven system design. The multimodal hypothesis, while well-motivated by successes in other document understanding tasks, does not hold for the receipt matching domain. As pre-trained language models continue to improve and become more efficient, we anticipate that text-based contrastive retrieval will remain the practical foundation for receipt entity resolution, with multimodal extensions reserved for domains where visual diversity provides a genuine discriminative signal.

References

- Appalaraju, S., Jasani, B., Kota, B. U., Xie, Y., & Manmatha, R. (2021). DocFormer: End-to-End Transformer for Document Understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 993–1003.
- Baltrusaitis, T., Ahuja, C., & Morency, L.-P. (2019). Multimodal Machine Learning: A Survey and Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2), 423–443.
- Brunner, U., & Stockinger, K. (2020). Entity Matching with Transformer Architectures — A Step Forward in Data Integration. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, pp. 463–473.
- Guo, H., Tang, J., Zeng, W., Zhao, X., & Liu, L. (2021). Multi-modal Entity Alignment in Hyperbolic Space. *Neurocomputing*, 461, 598–607.
- Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 1597–1607.
- Christophides, V., Efthymiou, V., Palpanas, T., Papadakis, G., & Stefanidis, K. (2021). An Overview of End-to-End Entity Resolution. *ACM Computing Surveys*, 54(6), Article 127.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (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 (NAACL-HLT)*, pp. 4171–4186.

- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Ebraheem, M., Thirumuruganathan, S., Joty, S., Ouzzani, M., & Tang, N. (2018). Distributed Representations of Tuples for Entity Resolution. *Proceedings of the VLDB Endowment*, 11(11), 1454–1467.
- Elmagarmid, A. K., Ipeirotis, P. G., & Verykios, V. S. (2007). Duplicate Record Detection: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1), 1–16.
- Fellegi, I. P., & Sunter, A. B. (1969). A Theory for Record Linkage. *Journal of the American Statistical Association*, 64(328), 1183–1210.
- Huang, Y., Lv, T., Cui, L., Lu, Y., & Wei, F. (2022). LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22)*, pp. 4083–4091.
- Huang, Z., Chen, K., He, J., Bai, X., Karatzas, D., Lu, S., & Jawahar, C. V. (2019). ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1516–1520.
- Johnson, J., Douze, M., & Jégou, H. (2019). Billion-Scale Similarity Search with GPUs. *IEEE Transactions on Big Data*, 7(3), 535–547.
- Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W.-t. (2020). Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781.
- Köpcke, H., Thor, A., & Rahm, E. (2010). Evaluation of Entity Resolution Approaches on Real-World Match Problems. *Proceedings of the VLDB Endowment*, 3(1-2), 484–493.
- Koutsikakis, J., Chalkidis, I., Malakasiotis, P., & Androutsopoulos, I. (2020). GREEK-BERT: The Greeks Visiting Sesame Street. In *Proceedings of the 11th Hellenic Conference on Artificial Intelligence (SETN)*, pp. 110–117.
- Le-Khac, P. H., Healy, G., & Smeaton, A. F. (2020). Contrastive Representation Learning: A Framework and Review. *IEEE Access*, 8, 193907–193934.

- Li, P., Gu, J., Kuen, J., Morariu, V. I., Zhao, H., Jain, R., Manjunatha, V., & Liu, H. (2021). SelfDoc: Self-Supervised Document Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5652–5660.
- Li, J., Xu, Y., Lv, T., Cui, L., Zhang, C., & Wei, F. (2022a). DiT: Self-supervised Pre-training for Document Image Transformer. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22)*, pp. 3530–3539.
- Li, Y., Li, J., Suhara, Y., Doan, A., & Tan, W.-C. (2020). Deep Entity Matching with Pre-Trained Language Models. *Proceedings of the VLDB Endowment*, 14(1), 50–60.
- Loshchilov, I., & Hutter, F. (2019). Decoupled Weight Decay Regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Karahalios, G., & Vigfusson, Y. (2018). Deep Learning for Entity Matching: A Design Space Exploration. In *Proceedings of the 2018 International Conference on Management of Data (SIGMOD)*, pp. 19–34.
- Oord, A. van den, Li, Y., & Vinyals, O. (2018). Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*.
- Outsios, S., Karatsalos, C., Skianis, K., & Vazirgiannis, M. (2020). Evaluation of Greek Word Embeddings. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC)*, pp. 2543–2551.
- Papadakis, G., Ioannou, E., Thanos, E., & Palpanas, T. (2021). The Four Generations of Entity Resolution. *Synthesis Lectures on Data Management*, Morgan & Claypool.
- Park, S., Shin, S., Lee, B., Lee, J., Surh, J., Seo, M., & Lee, H. (2019). CORD: A Consolidated Receipt Dataset for Post-OCR Parsing. In *Workshop on Document Intelligence at NeurIPS 2019*.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pp. 8024–8035.
- Peeters, R., Steiner, A., & Bizer, C. (2025). Entity Matching using Large Language Models. In *Proceedings of the 28th International Conference on Extending Database Technology (EDBT)*.

- Peeters, R., Bizer, C., & Glavaš, G. (2024). Cross-Lingual Learning Strategies for Improving Product Matching Quality. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing (SAC)*, pp. 560–569.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 8748–8763.
- Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3982–3992.
- Robertson, S., & Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4), 333–389.
- Robinson, J., Chuang, C.-Y., Sra, S., & Jegelka, S. (2021). Contrastive Learning with Hard Negative Samples. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Salton, G., & Buckley, C. (1988). Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, 24(5), 513–523.
- Shah, K., Kopru, S., & Ruvini, J.-D. (2018). Neural Network Based Extreme Classification and Similarity Models for Product Matching. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 8–15.
- Tracz, J., Wójcik, P. I., Jasinska-Kobus, K., Belluzzo, R., Mroczkowski, R., & Gawlik, I. (2020). BERT-based Similarity Learning for Product Matching. In *Proceedings of the Workshop on Natural Language Processing in E-Commerce (ECNLP)*, pp. 66–75.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., &

- Polosukhin, I. (2017). Attention Is All You Need. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pp. 5998–6008.
- Wang, D., Raman, B., Perot, V., Sarikaya, R., Agichtein, E., & Halevy, A. (2024). DocLLM: A Layout-Aware Generative Language Model for Multimodal Document Understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 8440–8465.
- Wang, L., Yang, N., Huang, X., Jiao, B., Yang, L., Jiang, D., Majumder, R., & Wei, F. (2022). Text Embeddings by Weakly-Supervised Contrastive Pre-training. *arXiv preprint arXiv:2212.03533*.
- Wang, P., Zeng, X., Chen, L., Ye, F., Mao, Y., Zhu, J., & Gao, Y. (2023). PromptEM: Prompt-tuning for Low-Resource Generalized Entity Matching. *Proceedings of the VLDB Endowment*, 16(2), 369–378.
- Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020). MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 5776–5788.
- Wilke, M., & Rahm, E. (2021). Towards Multi-Modal Entity Resolution for Product Matching. In *GI-Workshop on Foundations of Databases (GvDB)*, CEUR Workshop Proceedings, Vol. 3075.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., & Rush, A. M. (2020). Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, pp. 38–45.
- Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., & Zhou, M. (2020). LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 1192–1200.
- Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florêncio, D., Zhang, C., Che, W., Zhang, M., & Zhou, L. (2021). LayoutLMv2: Multi-modal Pre-training for Visually-Rich Document Understanding. In *Proceedings of the 59th Annual Meeting of the Association for*

Computational Linguistics (ACL), pp. 2579–2591.

Yang, L., Wang, Q., Yu, Z., Doan, A., & Halevy, A. (2022). MAVE: A Product Dataset for Multi-source Attribute Value Extraction. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 1256–1265.

Appendix A: Greek Text Normalization

The following Python functions implement the Greek text normalization pipeline, ported from the production system’s PHP implementation.

9.6 A.1 OCR Text Sanitization

```
import re

def ocr_text_sanitized(text: str) -> str:
    """
    Port of ReceiptItem::ocrTextSanitized() from
    snap-api/app/Models/ReceiptItem.php:144-173

    Cleans raw OCR text by removing special characters,
    normalizing whitespace, and removing Greek accents.
    """
    # Remove non-alphanumeric except Greek letters, digits, and spaces
    text = re.sub(
        r'[^Α-Ζ0-9\u0370-\u03FF\u1F00-\u1FFF+ ]', ' ',
        text, flags=re.IGNORECASE
    )
    # Remove additional special characters
    text = re.sub(
        r'[\~:}{!&+"'\[\]\_()\|\\\/]', ' ',
        text, flags=re.IGNORECASE
```

```
)

# Fix percentage patterns: "50%ΛΙΠΑΡΑ" → "50% ΛΙΠΑΡΑ"
text = re.sub(
    r'(\w) *%(\w)', r'\1% \2',
    text, flags=re.IGNORECASE
)

# Fix negative numbers: "ΕΚΠΤ- 2" → "ΕΚΠΤ -2"
text = re.sub(
    r'(\w|\s)- *(\d)', r'\1 -\2',
    text, flags=re.IGNORECASE
)

# Preserve decimal numbers: "3 . 5" → "3.5"
text = re.sub(
    r'(\d+)\s*\.\s*(\d+)', r'\1.\2',
    text, flags=re.IGNORECASE
)

# Remove Greek accents
ACCENTED = 'ΑΕΗΙΪΘΥΨΩάέήίϊίóóύύώ'
UNACCENTED = 'AEHIIOYYΩαεηιλλιοουσσω'
trans = str.maketrans(ACCENTED, UNACCENTED)
text = text.translate(trans)

# Normalize whitespace

return ' '.join(text.split()).strip()
```

9.7 A.2 Greek Normalization

```
def greek_normalize(text: str) -> str:
    """
    Port of StringMixin::greekNormalize() from
    vault-api/app/Mixins/StringMixin.php:78-91

    Full normalization: lowercase, remove accents,
    normalize final sigma, remove special characters.
```

"""

```
text = text.lower()

# Keep only alphanumeric and Greek characters
text = re.sub(
    r'[^a-z0-9\u0370-\u03FF\u1F00-\u1FFF+ ]', ' ',
    text, flags=re.IGNORECASE
)

text = text.strip()

# Remove accents and normalize final sigma
ACCENTED = 'άέήίϊίίόούύώς'
UNACCENTED = 'αεηιλλουσσωσ'

trans = str.maketrans(ACCENTED, UNACCENTED)

return text.translate(trans)
```

9.8 A.3 Verification

The normalization functions were verified by comparing their output against the production PHP implementation on a sample of 100 receipt items. All outputs matched exactly.

Appendix B: Detailed Results Tables

Table B.1: Full per-experiment metrics with 95% bootstrap confidence intervals

Experiment	MRR	95% CI
B1 (TF-IDF)	0.8126	[0.810, 0.815]
B2 (BM25)	0.7070	[0.704, 0.710]
B3 (OpenAI API)	0.7683	[0.766, 0.771]
T1 (GreekBERT)	0.9834	[0.9827, 0.9841]
T2 (mMiniLM)	0.9761	[0.9753, 0.9770]

Note: Multimodal experiments (M1–M3, L1) were not conducted because the image probe

showed no benefit from visual features (see Section 7.1.3). Bootstrap confidence intervals are computed from 1,000 resamples of the test set.

Appendix C: Qualitative Examples

9.9 C.1 Hard Items Where Neural Models Succeed

Since the image probe showed no benefit from visual features, multimodal models were not trained, and no multimodal-vs-text-only comparison examples exist. Instead, we present examples of items that the best neural model (T1, GreekBERT) resolves correctly but that statistical baselines fail on, illustrating the value of contextual text embeddings.

Example 1: Brand + volume disambiguation - OCR Text: “ΓΑΛΛΑ ΦΡΕΣΚΟ ΔΕΛΤΑ 1.5LT”

- True Product: “γαλα φρεσκο δελτα πληρες 1,5lt” - T1 (GreekBERT) prediction: correct (rank 1) - B1 (TF-IDF) prediction: “γαλα φρεσκο ολυμπος 1lt” (incorrect — matched on common tokens, confused volume and brand) - Analysis: The contrastive model learns that brand+volume together form a discriminative pattern, while TF-IDF treats each token independently and is misled by high-frequency terms like “γαλα” and “φρεσκο”. The catalog contains dozens of milk products, making this a challenging disambiguation case.

Example 2: Two-letter abbreviation decoding - OCR Text: “ΜΙΙΥΡ.ΑΜΣΤΕΛ FR 6X330”

- True Product: “μπιρα amstel free κουτι 6x330ml” - T1 (GreekBERT) prediction: correct (rank 1) - B1 (TF-IDF) prediction: “μπιρα amstel premium lager 330ml” (incorrect — failed to resolve abbreviation “FR” as “free”) - Analysis: Fine-tuned GreekBERT learns receipt abbreviation conventions (FR=free, 6X330=multipack) from training pairs, whereas TF-IDF has no mechanism to learn that “FR” maps to “free” rather than “French” or any other expansion.

Example 3: Multi-level abbreviation - OCR Text: “ΤΥΡ.ΦΕΤΑ ΠΟΠ ΔΩΔ.400G” - True

Product: “τυρι φετα π.ο.π. δωδωνη 400γρ” - T1 (GreekBERT) prediction: correct (rank 1) - B1 (TF-IDF) prediction: “τυρι φετα βαρελιστα χωριο 400γρ” (incorrect — failed to resolve abbreviations) - Analysis: The receipt text is heavily abbreviated (“ΤΥΡ.”=“τυρι”, “ΔΩΔ.”=“δωδωνη”, “ΠΟΠ”=“π.ο.π.”). Contrastive training on receipt-product pairs allows

GreekBERT to implicitly learn these abbreviation mappings, while character-level overlap in TF-IDF is insufficient.

Example 4: Category term replacement - OCR Text: “DIGESTIVE BAR Κ.ΦΡΟΥΤ & ΣΟΚΟΛ 28ΓΡ 5ΤΕΜ” - True Product: “Μπάρες δημητριακών ΠΑΠΑΔΟΠΟΥΛΟΥ Digestive κόκκινα φρούτα & σοκολάτα γαλ. 5x28gr” - T1 (GreekBERT) prediction: correct (rank 1) - B1 (TF-IDF) prediction: “Μπισκότα ΠΑΠΑΔΟΠΟΥΛΟΥ Digestive 250gr” (incorrect — wrong product type) - Analysis: The receipt omits the brand “ΠΑΠΑΔΟΠΟΥΛΟΥ” entirely and abbreviates “κόκκινα φρούτα” to “Κ.ΦΡΟΥΤ” and “σοκολάτα” to “ΣΟΚΟΛ”. GreekBERT resolves these through learned sub-word associations. TF-IDF latches onto “Digestive” (the only shared full token) and returns the wrong product.

Example 5: Greek-Latin script bridging - OCR Text: “ΚΡΕΜΑ ΞΑΝ Χ.ΓΛ ΒΑΛΣΑΜΙΚΗ 500ML” - True Product: “Κρέμα βαλσάμικο ΞΑΝ 500ml” - T1 (GreekBERT) prediction: correct (rank 1) - B1 (TF-IDF) prediction: “Κρέμα γάλακτος ΞΑΝ 200ml” (incorrect — wrong product) - Analysis: The receipt adds “Χ.ΓΛ” (χωρίς γλουτένη, gluten-free) and uses a different word form “ΒΑΛΣΑΜΙΚΗ” (feminine adjective) vs. “βαλσάμικο” (neuter). TF-IDF is confused by the extra tokens and the morphological mismatch.

Example 6: Word reordering - OCR Text: “ΜΠΟΥΓΑΤΣΑ ΚΡΕΜΑ ΜΑΡΑΤΑ 450ΓΡ” - True Product: “Μπουγάτσα ΜΑΡΑΤΑ με κρέμα 450gr” - T1 (GreekBERT) prediction: correct (rank 1) - B1 (TF-IDF) prediction: correct (rank 1) — but with a lower confidence score (0.68 vs. 0.92) - Analysis: Both models handle this word reordering, but GreekBERT’s contextualized representations are order-insensitive (via self-attention), while TF-IDF’s n-grams are inherently position-independent. The difference in confidence score reflects GreekBERT’s deeper understanding of the text semantics.

Example 7: Origin generalization - OCR Text: “ΠΑΤΑΤΕΣ ΑΙΓΥΠΤΟΥ.” - True Product: “Πατάτες εισαγωγής (μαναβική) - τιμή κιλού” - T1 (GreekBERT) prediction: correct (rank 1) - B1 (TF-IDF) prediction: “Πατάτες εγχώριες χύμα (μαναβική) – τιμή κιλού” (incorrect — wrong origin) - Analysis: “ΑΙΓΥΠΤΟΥ” (from Egypt) must be mapped to the generic catalog term “εισαγωγής” (imported). This requires understanding that Egypt is a foreign country, hence the product is imported. TF-IDF matches on “ΠΑΤΑΤΕΣ” and returns the domestic variant, which has higher n-gram overlap with common receipt patterns.

Example 8: Extreme slash abbreviation - OCR Text: “APT/ΣΜΑ ΚΑΡΑΜΟΛΕΓΚΟΣ ΨΙΧΑ ΣΤΑΠΕΝΙΟ 480G” - True Product: “Ψίχα τοστ ΚΑΡΑΜΟΛΕΓΚΟΣ σταρένιο 480gr” - T1 (GreekBERT) prediction: correct (rank 1) - B1 (TF-IDF) prediction: “Ψωμί ΚΑΡΑΜΟΛΕΓΚΟΣ αργοψημένο σταρένιο 750gr” (incorrect — wrong sub-brand) - Analysis: “APT/ΣΜΑ” (αρτοσκεύασμα) is a general bread term, while the product is specifically “Ψίχα τοστ”. GreekBERT learns that “ΨΙΧΑ” in the receipt is the distinguishing token and uses it to select the correct product, while TF-IDF focuses on “ΚΑΡΑΜΟΛΕΓΚΟΣ” + “ΣΤΑΠΕΝΙΟ” and returns the wrong bread variety.

9.10 C.2 Failure Cases

Representative failure cases of the best model (T1, GreekBERT), classified by error category. Understanding these failures is important both for assessing the model’s true capabilities and for identifying areas where further improvement is possible.

Example 1: Size variant confusion - OCR Text: “μερεντα με φουντουκι 360g” - True Product: “πραλινα φουντουκιου merenda 360gr” - Model prediction: “μερεντα πραλινα φουντουκιου 400gr” (incorrect — wrong size) - Error category: Language/transliteration mismatch + size disambiguation - Analysis: The receipt uses the Greek transliteration “μερεντα” while the catalog stores the brand in Latin script as “merenda”. The model correctly identifies the brand and product type but selects the wrong package size (400gr instead of 360gr). This is the most common type of neural model error: a “near miss” where the model understands the product identity but cannot precisely resolve the size attribute from abbreviated text.

Example 2: Dialectal variation and mixed script - OCR Text: “κολοκυθια εγχωρια” - True Product: “κολοκυθακια ελληνικα” - Model prediction: “κολοκυθια εισαγωγησ” (incorrect — wrong origin) - Error category: Dialectal variation + mixed script - Analysis: The receipt mixes Greek and Latin characters (“κολοκυθια” with Latin “ia” instead of Greek “ia”) and uses the informal “εγχωρια” (domestic) instead of the catalog term “ελληνικα” (Greek). The mixed script disrupts tokenization, and the synonym gap between “εγχωρια” and “ελληνικα” is not bridged by the model. Interestingly, the model returns the imported variant of the same vegetable — it understands the product but confuses the origin.

Example 3: Ground truth annotation error - OCR Text: “μπίρα amstel free κουτί 6x330ml” - True Product: “λεμόνια” (annotated ground truth) - Model prediction: “μπιρα amstel free

κουτι 6x330ml” (semantically correct, marked wrong) - Error category: Ground truth error - Analysis: The ground truth label is clearly wrong — a beer item cannot correspond to lemons. This is a data annotation error in the original mapping. The model’s prediction is in fact correct, but it is penalized by the erroneous ground truth. Such cases inflate the reported error rate and highlight the importance of data quality auditing. We estimate approximately 15% of “errors” fall into this category.

Example 4: Extreme abbreviation with insufficient context - OCR Text: “ΓΑΛ 1LT” - True Product: “Γάλα φρέσκο ΟΛΥΜΠΙΟΣ πλήρες 3.5% 1lt” - Model prediction: “Γάλα φρέσκο ΔΕΛΤΑ πλήρες 3.5% 1lt” (incorrect — wrong brand) - Error category: Insufficient information in receipt text - Analysis: The OCR text “ΓΑΛ 1LT” contains only the abbreviated word for milk and the volume — no brand, no fat percentage, no subcategory. The catalog contains over 20 different 1-liter milk products from different brands. Without additional context (e.g., the store’s product range, or the price), even a human expert would struggle to determine the correct brand. The model makes a reasonable guess but picks the wrong brand.

Example 5: Flavor variant confusion - OCR Text: “ΦΡΥΓΑΝΙΕΣ ELITE ΣΙΤΟΥ 250ΓΡ” - True Product: “Φρυγανιές ELITE σταρένιες 250gr” - Model prediction: “Φρυγανιές ELITE ολικής αλέσεως 250gr” (incorrect — wrong variant) - Error category: Synonym confusion at the variant level - Analysis: “ΣΙΤΟΥ” (wheat) and “σταρένιες” (made from wheat flour) are closely related concepts, but the receipt uses a different term than the catalog. The model retrieves a different variant (whole wheat instead of regular wheat) — again a “near miss” error that reflects the fine-grained distinction between product variants that even share the same base ingredient.

Example 6: Brand name collision - OCR Text: “ΚΡΕΜΜΥΔΙΑ ΞΕΡΑ ΕΓΧΩΡΙΑ.” - True Product: “Κρεμμύδια ξερά ελληνικά ξανθά (μαναβική) – τιμή κιλού” - Model prediction: “Κρεμμύδια ξερά ελληνικά κόκκινα (μαναβική) – τιμή κιλού” (incorrect — wrong color) - Error category: Missing distinguishing attribute - Analysis: The receipt text does not specify the onion color (ξανθά = blonde/yellow vs. κόκκινα = red), and the dialectal “ΕΓΧΩΡΙΑ” → “ελληνικά” mapping is correctly resolved. The error is purely due to the missing color attribute: the model must guess between yellow and red onions with no textual signal to guide the decision.

9.11 C.3 Image Feature Visualization

Attention heatmap visualization was not pursued. Since the image probe experiment (Section 7.1.3) demonstrated that receipt line-item images carry no discriminative signal beyond what is already captured by the text encoder, generating and analyzing attention heatmaps for the image encoder would not yield meaningful insights. The uniform monospaced font and lack of visual product differentiation in receipt images means there are no salient visual patterns for the encoder to attend to.

Author's Statement

I hereby expressly declare that, according to the article 8 of Law 1559/1986, this dissertation is solely the product of my personal work, does not infringe any intellectual property, personality and personal data rights of third parties, does not contain works/contributions from third parties for which the permission of the authors/beneficiaries is required, is not the product of partial or total plagiarism, and that the sources used are limited to the literature references alone and meet the rules of scientific citations.