VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Deep Learning and Its Applications (CO3133)

Assignment 1: Application of Deep Learning in Classification of Text

Advisor: Ph.D Lê Thành Sách
Students: Vũ Hoàng Tùng - 2252886
Vũ Minh Quân – 2212828
Nguyễn Việt Hoàng – 2252235

HO CHI MINH CITY, April 2026

Acknowledgement

We would like to show our foremost appreciation to PhD. Lê Thành Sách. His Lectures on Deep Learning model and its implementation always deliver fundamentals of great depth as well as illustration of practical applications by truly comprehensive means. His guidance and visualization of knowledge have laid foundation to our current and upcoming Assignments.

We also thank the creator of EURLEX57k that provide dataset and resources for our study, alongside the pioneers of Deep Learning researchers, especially in the field of Classification. Their intelligent and creative works truly motivate us to challenge ourself more out of the norm. Final thanks for any support and consultancy of our friends of same department.

Contents

Chapter 1

Text classification

Classifying long—form documents within a high-dimensional label space ($10^3 \rightarrow 10^6$) has been always a formiddable challenge, yet has been needed for solution more than ever, due to the non-stop accumulation of documents, papers of complex domain with many topic overlap. As such a field called XMC.

This study proposes the 2 LSTMbased and Transformerbased architectures specialized for XMC, benchmarks them on their ability to handle semantic overlap and document length in the Eurlex57k dataset.

1.1 Introduction

1.1.1 Motivation

The current era is witnessing an rapid technological development, characterized by millions of research entities and organizations contributing to a global knowledge base. Thanks to the invention in the digital era, these novel contributions are now preserved and accumulated into massive archives containing billions of documents, and being more accessible than ever on the Internet. Platforms such as Stack Overflow, ResearchGates, and various specialized academic and informative portals are built by researches organizations as primary access points for this vast data. On these platforms, tens of thousands of technical queries, documents, and research articles are uploaded daily, forming a diverse and ever-growing unstructured dataset. Effective organization of this unstructured data is always wanted as to:

- Content Discovery: Enabling researchers and students to find specific technical solutions and topics among billions of docs.
- Expert Routing: Automatically directing new queries to the most suitable contributors, platforms, ... ussing domain-based tags.
- Knowledge Graph Construction: Establishing semantic links between disparate academic topics to facilitate cross-disciplinary learning.
- E.c.t .....

However, traditional categorization methods often struggle with the inherent complexity of academic discourse. Scholarly texts are characterized by:

- Syntax's Complexity: The use of passive voice, nested clauses, and semantic expression that can only be inferred by humans, make traditional techniques such as Dependency Parsers, keyword mapping, ... fail.
- Entity Ambiguity: entities within academic texts appears most frequently as citations or cross-references rather than main focus entities, acting as distractors causing standard Named Entity Recognition (NER) systems to fail from capturing the true key concept.

From that modern Deep Learning classification models have significantly enhanced performance by capturing global context. However, basic classification with a few output head, with 1 class result won't cut it. The sheer presence of multi-topic documents, with potentially thousands of topics in each field, making the automation task a global difficult study area, that needs to be tackled to boost document searching, organizing and understanding publicly and internally within big companies.

1.1.2 Problem Statement

As long documents often cover multiple specialized subjects simultaneously, assigning a single category is no longer sufficient. The shortcomings have led research community to found a new field of study, Extreme Multi-label Classification (XMC) within the complex domain, which has then been with active leaderboards, datasets and contributions from many research teams and big org like Amazon, Google, Wiki, ...

The problem is simply explained as follows: given a complex input document x, the system

must identify a relevant subset of labels y from an extremely large and potentially hierarchical label space L. In the context of academic tagging, this task introduces several rigorous technical hurdles:

1. Extreme Scale: The number of potential tags ($|L|$) can reach hundreds of thousands, making traditional output layers computationally expensive.
2. Label Sparsity: Following a power-law distribution, a few "head" tags appear frequently, while the vast majority of specialized "tail" tags suffer from a severe lack of training data.
3. Multi-faceted Content: Academic documents often bridge multiple disciplines, requiring the model to understand complex label dependencies rather than treating categories as mutually exclusive.

This study focuses on leveraging advanced Deep Learning architectures to optimize tagging performance, specifically addressing the "long-tail" nature of academic data through specialized evaluation metrics such as nDCG@k and Propensity-scored F1.

1.2 Background and Literature Review

1.2.1 Background Knowledge

1.2.1.1 Theoretical Framework of Text Classification

Text classification is a fundamental task in Natural Language Processing (NLP) that involves assigning a document x to one or more categories from a predefined label set L. If x is expected to be mapped to only 1 label $l \in L$, it is called Single Classification, else if l is a list suffices $l \subseteq L$, it is called Multilabel Classification

Standard Classification

In traditional, common settings, such as Multi-class Classification, each document is mapped to exactly one label ($|y| = 1$). The model typically employs a Softmax activation layer at the output to generate a probability distribution where:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{|L|} e^{z_j}} \tag{1.1}$$

This approach assumes that labels are mutually exclusive, which is rarely the case in complex academic or legal contexts where a single entity may belong to multiple domains.

Extreme Multi-label Classification (XMC)

Extreme Multi-label Classification (XMC) extends this by allowing a document to be associated with a subset of labels $y \subseteq L$, where the size of L is "extreme" (ranging from 10^2 to 10^6). The challenges in XMC are:

- Computational Complexity: The output layer must handle a massive number of parameters, expanding matrixes both in, out, and between (feature matrixes). Even the most advanced Deep Learning model struggle to remember feature matrixes from all labels.
- Data Imbalance: Labels typically follow a Power-law distribution, where a few super common "head" labels have abundant data, while the rare "long-tail" labels suffer from extreme sparsity.

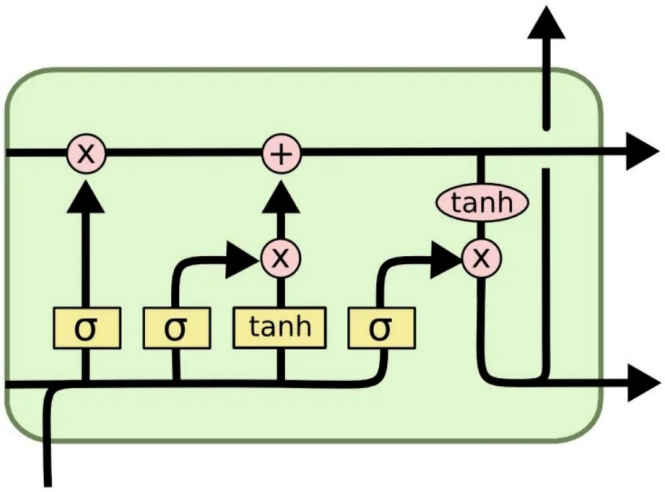


Figure 1.1: Architecture of LSTM

1.2.1.2 Deep Learning Architecture in Textual Modelling

Deep Learning Architectures in NLP context are model aiming to capture the semantic essence from sequence of tokens (words, texts).The effectiveness of text classification on complex academic data relies on the model's ability to capture both local sequential patterns and global semantic dependencies. This section discusses the two backbone architectures used in our study: LSTM and Transformers.

Long Short-Term Memory (LSTM)

Originally, Long Short-Term Memory (LSTM) networks were proposed by Hochreiter and Schmidhuber in 1997 to address the fundamental "vanishing gradient" problem inherent in standard Recurrent Neural Networks (RNNs). The core intuition behind LSTM is similar to the skip connections found in modern ResNet architectures; it establishes a "cell state" highway (C_t) that allows information to flow across long sequences with minimal or no interference at all. This flow is regulated by three specialized gating mechanisms: the forget gate (f_t), the input gate (i_t), and the output gate (o_t). Specifically, the forget gate acts as a learned residual weight, deciding which percentage of the previous knowledge should be preserved:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1.2}$$

The input gate then determines the magnitude of new information to be stored, and the cell state is updated through a summation process that ensures stable gradient propagation:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{1.3}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{1.4}$$

Transformer

The year 2017 marked a paradigm shift with the introduction of the fTransformer architecture in the paper "Attention Is All You Need." TThe core idea of the Transformer is simple: let the

model learn where to focus given a long sequences of paragraph. This is all thanks to the Multi-head Attention mechanism, which maps an input into a new space where tokens are represented by their context. The process involves three linear projections:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V \tag{1.5}$$

Where Q (Query) acts as a search request, K (Key) represents the characteristics of other tokens, and V (Value) contains the actual content. The **Attention Score** is calculated by the dot product of Q and K^T, representing the semantic similarity between tokens. To maintain numerical stability for large dimensions, we scale by $\sqrt{d_k}$:

$$\text{Score} = \frac{QK^T}{\sqrt{d_k}} \tag{1.6}$$

Finally, the **Softmax** function generates a probability distribution, ensuring that the final output is a weighted sum of V based on their relevance to the current context:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1.7}$$

This mechanism enables the model to capture global dependencies and complex academic grammar more robustly than sequential models.

1.2.1.3 Optimization Algorithm for XMC

The optimization algorithm is mechanism for model to improve themselves given sufficient data, via Loss function. In the landscape in XMC is uniquely challenging due to the massive label space and the dominance of negative samples. To achieve high precision across the entire label spectrum, we implement a multi-layered optimization strategy.

Binary Cross-Entropy (BCE)

Binary Cross-Entropy serves as the fundamental baseline for multi-label tasks by treating each label as an independent Bernoulli distribution. The objective is to minimize the logarithmic distance between the predicted probability $p_i = \sigma(z_i)$ and the ground truth $y_i \in \{0, 1\}$:

$$L_{BCE} = -\sum_{i=1}^{|L|}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)] \tag{1.8}$$

While effective for standard tasks, BCE's main weakness in XMC is its "gradient indifference"—it assigns equal importance to all samples, causing rare label loss to be a lot smaller, causing vanishment of gradient upon Deep Learning model training.

Asymmetric Loss (ASL)

To combat the imbalance between positive and negative labels, we utilize Asymmetric Loss (ASL). ASL operates on the intuition that negative samples, being the vast majority, should be suppressed more aggressively to allow the model to focus on hard samples. This is achieved by introducing asymmetric focusing parameters γ_- and γ_+, along with a probability shifting margin m:

$$L_{ASL} = \sum_{i=1}^{|L|} \begin{cases} (1 - p_i)^{\gamma_+} \log(p_i) & \text{if } y_i = 1 \\ (\max(0, p_i - m))^{\gamma_-} \log(1 - \max(0, p_i - m)) & \text{if } y_i = 0 \end{cases} \tag{1.9}$$

By setting $\gamma_- > \gamma_+$, the loss function "shrinks" the gradients of easy negatives. The margin m further "zeroes out" any contribution from negatives with very low predicted probabilities, effectively filtering out background noise.

Class-Balanced Weighting (CB-Weight)

While ASL handles sample-level difficulty, Class-Balanced (CB) Weighting addresses class-level frequency. The core idea is to model the "effective number" of samples, acknowledging that new data provides diminishing returns in information gain as the dataset grows. This is modeled via a strictly concave function:

$$W = \frac{1}{E} = f(n) = \frac{(1-\beta)}{(1-\beta^n)} \text{ s.t } \beta < 1 \tag{1.10}$$

With $\beta < 1$, this function ensures that the information gain of a sample decreases as $n \to \infty$ or say $n_{head} \gg n_{tail}$, which is mathematically expressed by this:

$$|f'(n_{tail})| > |f'(n_{head})| \quad \text{for} \quad n_{head} > n_{tail} \tag{1.11}$$

Integrating this weight W_i into the final objective $\mathcal{L}_{Total} = \frac{1}{|L|} \sum W_i \cdot L_{ASL}$ ensures that rare "tail" labels generate sufficient gradient signal to be learned effectively without being overshadowed by "head" categories.

AdamW Optimizer

To navigate the complex, non-convex loss surface generated by ASL and CB-Weight, we employ **AdamW**, an evolution of the Adam optimizer that decouples weight decay from the gradient update. This separation is critical for Transformer-based architectures to ensure that adaptive learning rates do not interfere with regularization. The optimization process follows four logical steps:

1. **First Moment Estimation (Momentum):**

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{1.12}$$

   Functions as **inertia** to smooth out noisy gradients and overcome local minima, ensuring a consistent update direction.

2. **Second Moment Estimation (Adaptivity):**

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{1.13}$$

   Tracks the **uncentered variance** of gradients to scale updates; parameters with rare, sparse gradients (small v_t) receive larger effective steps.

3. **Bias Correction:**

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{1.14}$$

   Corrects the initial bias toward zero, ensuring numerical stability and reliable performance from the first epoch.

4. **Adaptive Update with Decoupled Weight Decay:**

$$\theta_{t+1} = \theta_t - \eta \left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} + \lambda \theta_t \right) \tag{1.15}$$

Performs the final parameter shift. The term $\lambda\theta_t$ penalizes large weights directly to improve generalization, **s.t.** $\eta > 0$ and $\lambda \geq 0$ (where η is the learning rate and λ is the weight decay coefficient).

1.2.2 Literature Review

To address eXtreme Multi-label Text Classification (XMC), State-of-the-Art (SOTA) models aim to efficiently map complex text to massive, highly sparse, and long-tail label spaces [1, 2].

Probabilistic Label Trees (PLTs) and Linear Models

Early scalable approaches like Parabel [3], Bonsai [4], and Slice [5] rely on hierarchical label partitioning or scalable linear classifiers. However, their use of non-contextual TF-IDF representations severely limits their ability to capture deep semantic text patterns.

Label-Wise Attention Networks (LWANs)

To overcome limitations of TF-IDF or pure statistic method in general, researchers resorted to Transformer. However, with $L > 1000$, even long multihead selfattention architecture which usually encode texts into a vector with dimension $D = 2048$ is not effective. To further leverage transformer architecture for XMC, LWANs [6, 7] utilize deep learning by employing a distinct attention head for each label, dynamically focusing on relevant tokens. Despite their robust performance, vanilla LWANs ignore the inherent structural information of the label hierarchy.

Hybrid Architectures (AttentionXML and Beyond)

AttentionXML [8] pioneered the combination of PLTs with bidirectional RNNs and label-aware attention. Instead of training with all label at once, it learns the tag relationship and split training phase based on constructed semantic hierarchy. It successfully catalyzed a new wave of highly efficient, lightweight frameworks like InceptionXML [9] and MatchXML [10], which further reduce computational bottlenecks via synchronized negative sampling and efficient text-label matching.

Transformers and Graph-aware Methods

Recently, models have shifted to pre-trained Transformers. CascadeXML [11] rethought Transformers for end-to-end multi-resolution XMC training, achieving superior performance by exploiting deep contextualized embeddings. Furthermore, to tackle data scarcity (few/zero-shot labels), models like ECLARE [12] and GalaXC [13] incorporate Graph Neural Networks (GNNs). By encoding label graph correlations, they enable the representations of rare labels to directly benefit from frequent, nearby labels.

1.2.3 Conclusion

To address issues in XMC, many methods have been studied to fruitation of success. Despite that, even the SOTAs struggle to achive $P@3 > 0.8$. Inspired by previous work, this report recreate some of the fundamental techniques and paradigm across XMC researches, perform experiments

and explaining problems as well as solutions, optimization stratergies from both mathematical and empirical point of view.

1.3 In-depth Dataset Analysis: Eurlex57k

This section provides a comprehensive analysis of the **Eurlex57k** dataset, which serves as the primary benchmark for evaluating our XMC architecture. Understanding the linguistic and statistical properties of this legal corpus is critical to indentify dificulties and novelty, from then justify the choice of advanced Deep Learning components and config.

1.3.1 Dataset Statistics and Textual Complexity

We performed an extensive Exploratory Data Analysis (EDA) on both the training set (45,000 samples) and the test set (6,000 samples). First is the distributions in length of title, body, recitals and full text (title + body + recitals) summarized in Table 1.1.

Table 1.1: Detailed Statistical Comparison of Document Lengths (Tokens)

Dataset	Metric	full text (t_len)	Recital (r_len)	Main (m_len)	Title	Head
Train	Mean	351.19	318.00	182.09	33.19	43.19
	Median (50%)	273.00	237.00	87.00	31.00	41.00
Test	Mean	350.36	317.44	175.36	32.91	43.02
	Median (50%)	274.00	237.00	85.00	31.00	41.00

From our statistical analysis, we draw some conclusions:

- **Positive Skewness:** In both sets, the Mean is significantly higher than the Median (e.g., m_len Mean 182 vs Median 87). This indicates a *heavy-tailed distribution* where a small number of extremely long documents (up to 3,236 tokens) shift the average, necessitating models that can handle long-range dependencies.
- **Sectional Correlation:** The "Recital" section (r_len) accounts for approximately 90% of the total document length. The high standard deviation (≈ 263) across Recitals suggests that while some laws are concise, others contain vast legal preambles that are critical for semantic disambiguation.
- **Label Sparsity Stability:** The $tags\_count$ remains remarkably stable across splits. With a mean of 5.07 and a median of 5.00, the "label-per-document" density is consistent, though the complexity ranges from 1 to 26 labels per document.

Table 1.2: Label Density Distribution per Document

Split	Count	Mean	Std	Min	50%	75%	Max
Train	45,000	5.07	1.69	1.00	5.00	6.00	26.00
Test	6,000	5.05	1.73	1.00	5.00	6.00	19.00

The observation that 75% of documents have 6 labels or fewer, while the maximum reaches 26, confirms the **multi-label** nature of the task. This variability suggests that the model must not only predict the "Head" labels but also correctly identify multiple "Tail" labels that appear in these high-density legal statutes.

1.3.2 Label Distribution: The Long-tail and Zero-shot Challenge

The label space of Eurlex57k, shown in Table 1.3, exhibits a classic **Extreme Multi-label** characteristic, where a tiny fraction of labels dominates the majority of document instances. Our statistical analysis of the 4,108 unique EuroVoc tags reveals a severe **Power-law (Long-tail) distribution**:

- **Head Labels (Frequent):** 739 labels (approximately 17.6%) appear more than 50 times. These "Head" labels, such as Tag 1309 (501 samples) and Tag 3568 (488 samples), are well-represented and can be easily optimized by standard cross-entropy.
- **Few-shot Labels (Tail):** An overwhelming majority of 3,369 labels (80.38%) appear fewer than 50 times. Within this group, "Bottom" labels (e.g., Tags 2019, 4732, 4115) appear only once in the entire training set, providing almost no statistical signal for traditional classifiers.
- **Zero-shot Labels:** We identified 85 labels in the test set that were entirely absent during the training phase, posing an extreme challenge for inductive inference.

Table 1.3: Label Categorization Summary

Category	Criteria	Count	Percentage (%)
Normal (Head)	$n \geq 50$	739	17.6%
Few-shot (Tail)	$1 \leq n < 50$	3,369	80.38%
Zero-shot	$n = 0$ (Train)	85	2.02
Total Unique Classes		**4,193**	**100%**

To further illustrate this disparity, we analyze the frequency gap between the most dominant and the most obscure labels in the dataset:

Table 1.4: Frequency Contrast: Top 10 vs. Bottom 10 Labels

Top Label ID	Frequency	Bottom Label ID	Frequency
1309	501	2019	1
3568	488	4732	1
2635	386	4115	1
20	386	2440	1
1118	383	869	1
1605	380	2354	1
693	379	4879	1
1644	301	3250	1
161	295	2317	1
2300	293	1700	1

This distribution provides the empirical justification for our optimization suite. Standard Cross-Entropy would naturally bias the model toward the 132 Head labels, accounting for a mere 5.2% in the training set (3.14% if you account for both dataset). By integrating **CB-Weight**, we compensate for the 94.8% of Tail labels by amplifying their gradient signal based on the effective number of samples, ensuring the model does not become a simple majority-class predictor.

1.3.3 Correlation and Semantic Hierarchy Analysis

Eurlex57k is grounded in the formal **EuroVoc taxonomy**, which expert-defined categories like Politics or Entertaiment. Though helpful, those are still limited to capture the more granular, latent semantic relationships of leaf tags, especially in our focus optimization of tail tags. To investigate this, we conducted a statistical correlation analysis to determine if labels could be statistically grouped into meaningful clusters.

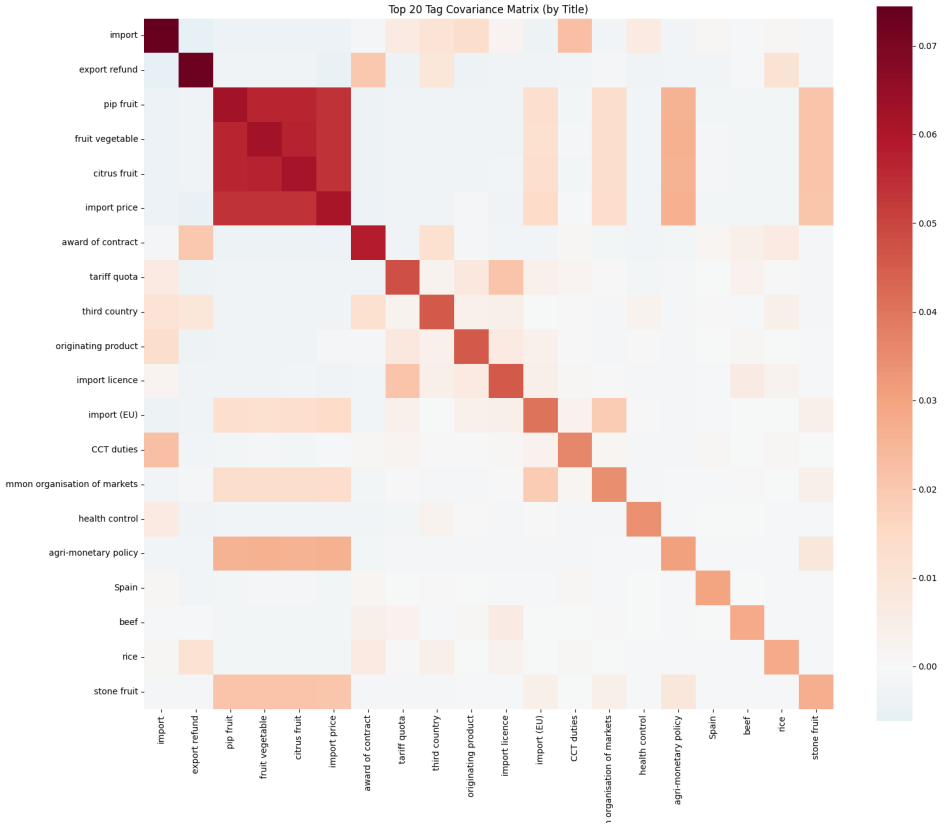


Figure 1.2: Covariance Matrix of Top 20 Tags: High density among "Head" labels indicates strong semantic clusters in common legal domains.

The initial analysis of high-frequency "Head" labels (Figure 1.2) suggests a promising structure. For instance, tags related to "Fruits and Vegetables" (citrus fruit, pip fruit, etc.) exhibit near-perfect correlation, forming a dense, localized semantic block. This suggests that for common, high-frequency domains, statistical grouping aligns well with expert hierarchies. A

Further experiment with intracommunity correlation (Figure 1.3) reveals a dual nature in the statistical clustering. On one hand, the diagonal—representing the relationship between the community head and individual tags—is noticeably darker than the surrounding cells. This is a positive indicator of thematic alignment, suggesting that these labels do share a common legal

domain.

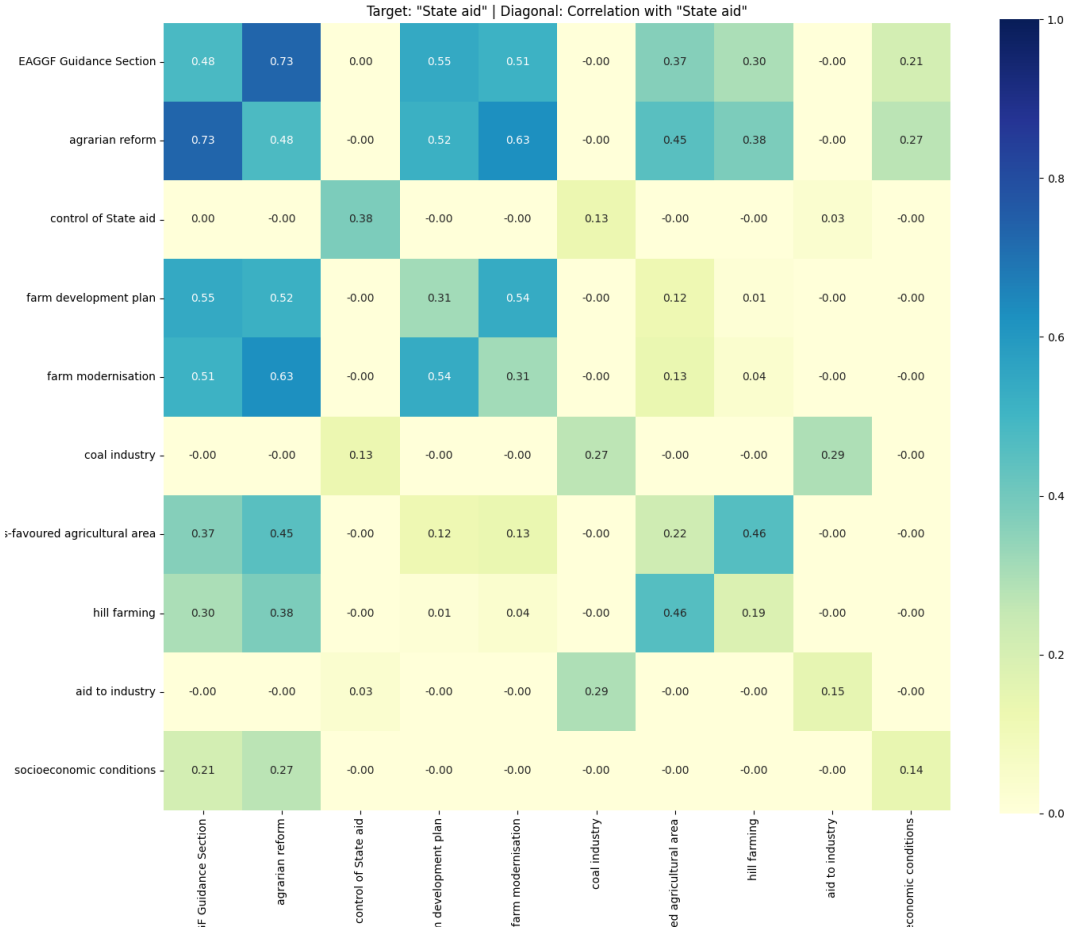


Figure 1.3: Correlation of community head "State aid" with its top 10 community members. The strong diagonal suggests a valid thematic grouping, yet the high-frequency inner clusters reveal underlying statistical bias.

However, a significant drawback emerges in the top three tags, which exhibit an intense inner correlation, not so ideal for clustering. This is likely because of their high frequency rather than precise legal synergy, indicating a statistical bias. When expanding this approach to the broader dataset using the Louvain community detection algorithm, these limitations become even more evident:

- **Hierarchical Complexity:** We identified major root nodes (e.g., Tag 4381 with over 1,000 mentions) acting as semantic backbones. However, lower-level tags often relate to multiple "parent candidates" (e.g., Tag 1026 linked to 1048, 2300, and 4271), creating a **Directed Acyclic Graph (DAG)** rather than a simple tree structure.
- **Failure on the Long-tail:** Approximately 1,577 out of 4,108 tags (over 38%) could not be assigned to any stable community.

While Louvain clustering is effective for "Head" tags with abundant co-occurrence data, it struggles to account for the $1.5k$ isolated and "Tail" tags. These labels suffer from such extreme data scarcity that they appear as "semantic noise" to traditional clustering algorithms.

This proves that treating tags as either independent categories (as in BCE) or as part of a rigid, simple cluster is insufficient. To succeed in XMC, the model must go beyond surface-level statistics and learn **Deep Label Dependencies** that can bridge the gap between frequent clusters and isolated tail entities.

1.3.4 Conclusion from EDA

The analysis of Eurlex57k confirms its position as one of the representative benchmarks in the Extreme Multi-label Classification (XMC) domain, accounting for all possible challenges in the field of XMC. The difficulty arises not only from the scale of the label space but also from the intrinsic nature of legal documentation.

Structural Complexity of Recitals: As evidenced by our statistical analysis, the "Recital" section accounts for over 90% of the document's total length, with some reaching over 3,000 tokens. These sections contain the legal justifications and precedents that are often more semantically dense than the "Main" articles themselves. Standard NLP models limited to 512 tokens would inherently lose nearly 80% of this critical context, necessitating advanced sequence modeling to capture the full legal reasoning. Moreover, as discussed in Introduction (Subsection 1.1.2), complex citations and references may act as noise in many sequentail model training.

Nature of Legal Language: In the legal domain, unlike normal writing, language is characterized by highly formalized prose, extensive use of passive voice, and a dense network of cross-references. Unlike general community forums (e.g., StackExchange), legal entities often function as citations rather than simple subjects. This creates a high level of semantic ambiguity where the model must distinguish between a statute being mentioned as a reference and a statute being the primary subject of the document. Different semantics, weird tokens symbolizing different technical citations and Laws code, all can ruin any general, performative Bert if are not handled well

Extreme Label-to-Document Ratio: With approximately 4,108 unique labels assigned to a corpus of 45,000 documents, the average label-to-document ratio is exceptionally high compared to standard classification tasks. Specifically, while the mean number of tags per document is approximately 5.07, the long-tail distribution reveals that 80% of the labels are "few-shot," occurring in fewer than 50 documents. This means that for nearly 3,400 specialized legal categories, the model has less than 50 or 0.1% of the total data for training, making it a true "Extreme" classification problem where missing a single rare tag could lead to significant failures in information retrieval and classification.

1.4 Method

The proposed framework operates as a cohesive pipeline, transforming raw legal documents into multi-label predictions through four distinct stages: Preprocessing, Encoding, Attention-based Classification, and Asymmetric Optimization.

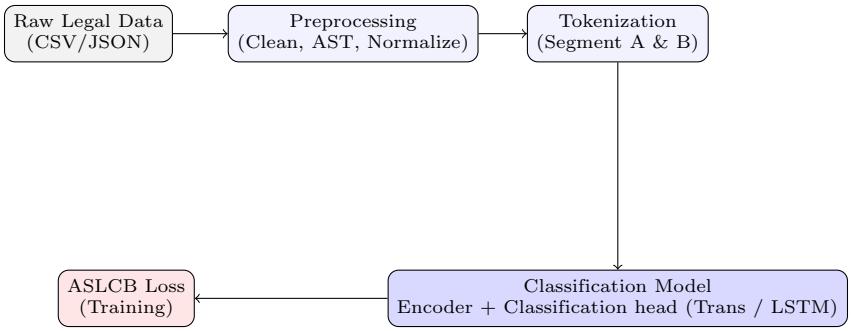


Figure 1.4: End-to-End Workflow: From raw legal text preprocessing to asymmetric loss optimization.

1. Data Preprocessing and Tokenization: Raw documents from the train_raw dataset undergo a specialized cleaning pipeline (as detailed in Section 3.2). The cleaned text is then partitioned into two functional segments: Segment A (Title + Body) and Segment B (Recitals). These segments are passed through the AutoTokenizer, converting words into numerical input_ids and attention_mask tensors, each capped at 512 tokens.

2. Feature Extraction (Backbone): The train_dataloader feeds these tensors into the Pretrained Transformer Backbone (all-MiniLM-L12-v2). The model processes both segments independently to extract the last_hidden_state, capturing deep contextual embeddings for every token in the 1024-token span.

3. Modular Feature Fusion and Attention: The hidden states are merged via the Multi-segment Feature Fusion layer (1024×384). This combined matrix is then ingested by the Label-Wise Attention (LWA) head. Each of the 4,108 labels performs a targeted search across the fused features to calculate the probability of its presence in the document.

4. Training vs. Inference Mode:

- Training Path: The output logits are fed into the ASLCB Loss function. This loss utilizes Class-Balanced weights ($\beta = 0.999$) and Asymmetric Focusing to penalize the model based on the extreme label imbalance, guiding the gradients to prioritize difficult "Tail" labels.
- Inference Path: During testing, the model generates sigmoid probabilities. These are filtered through an optimized global Threshold (e.g., 0.3) to produce the final set of predicted EuroVoc concepts, which are then validated against the ground truth using Micro/Macro F1 metrics.

1.4.1 Preprocessing

To ensure the high-quality input required for deep learning architectures, we implement a specialized preprocessing pipeline corresponding to the unique linguistic pattern of European Union legal documents. The raw Eurlex57k data contains structured string lists and legal-specific noise that can impede the learning process. Our preprocessing pipeline consists of several key stages:

- Structural Reconstruction: We utilize Abstract Syntax Tree (AST) parsing to convert string-represented lists (e.g., Recitals and Main Body) into continuous text streams, preserving the narrative flow of legal reasoning.
- Noise Reduction: Footnote indices and citations (e.g., `[1]`, `(1)`) are removed using regular expressions: `r'\[\s*\d+\s*\]'` and `r'\(\s*\d+\s*\)'`. Residual HTML tags are also stripped to clean web-crawled artifacts.
- Normalization: We apply NFC Unicode normalization and standardize special punctuation (e.g., converting stylized dashes '–', '—' and quotes '"', '"' to ASCII equivalents). This reduces vocabulary sparsity and ensures consistent embedding lookups.
- Standardization: Finally, the corpus undergoes whitespace collapse and case folding (lowercase conversion) to minimize the unique token count and improve generalization across similar legal terms.

Table 1.5: Comparison between Raw and Processed Legal Data (Sample ID: 32011D0690)

Feature	Raw Data (Input)	Processed Data (Output)
Text Casing	Mixed case (e.g., "Commission", "Article 29")	Lowercase (e.g., "commission", "article 29")
Footnotes	Contains markers: "...91/664/EEC (1)... Decision 2011/163/EU (2)..."	Markers removed: "...91/664/eec ... decision 2011/163/eu ..."
Unicode	Specialized characters: "14 October", "C(2011)"	Normalized: "14 october", "c 7167"
Structure	Nested lists as strings: `['The Annex...', 'This Decision...']`	Flattened narrative: "the annex to... this decision shall..."
Noise	Includes HTML-like spacing and special punctuation (—, ")	Stripped of tags and standardized punctuation (-)
Tags	EuroVoc numeric strings	Filtered and binarized tags (e.g., L1: 13, L1_Name: 4381)

Table 1.5 shows the text after being preprocessed. Key Observations from Preparation:

- **Noise Filtering:** By removing footnote markers like `(1)` or `[2]`, we prevent the model from assigning false importance to citation indices that do not carry semantic meaning.
- **Semantic Consolidation:** Converting all text to lowercase and normalizing Unicode characters ensures that the embedding layer does not create separate vectors for the same word (e.g., "Decision" vs "decision").
- **Label Mapping:** The raw EuroVoc concepts are mapped to a filtered subset (e.g., Level 1 categories), reducing the label space to a manageable yet representative set for extreme multi-label classification.

1.4.2 Deep Learning Architecture

To effectively classify the long-form and high-dimensional Eurlex57k dataset, we adopt a modular "Encoder-Head" design. In this framework, a pre-trained encoder extracts semantic features, which are then processed by specialized classification heads to predict EuroVoc labels.

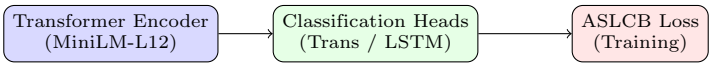


Figure 1.5: Architecture of Deep Learning Classification Model

1.4.2.1 Pre-trained Model: All-MiniLM-L12-v2

For the Encoder, we utilizes all-MiniLM-L12-v2, a distilled and highly efficient Transformer model, as its primary feature extractor. MiniLM follows the standard Transformer Encoder-only architecture but is optimized through a specialized distillation process. It consists of $L = 12$ Transformer blocks, each containing a Multi-Head Self-Attention (MHSA) layer and a Feed-Forward Network (FFN).

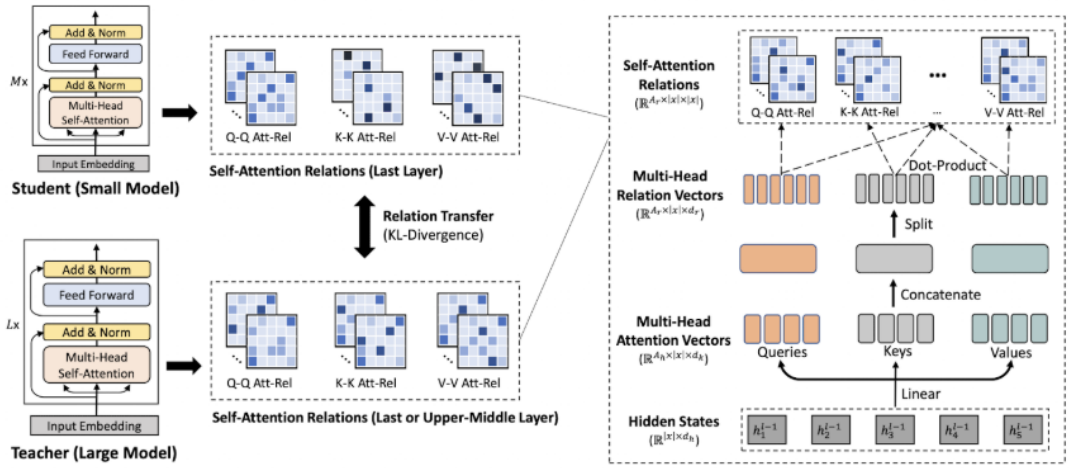


Figure 1.6: Distilation Paradigm of MiniLM v2 in general

Designed for sentence-level embeddings, this model transforms raw input tokens into a dense, 384-dimensional continuous vector space. Each token t_i is mapped to a hidden state $h_i \in \mathbb{R}^{384}$:

$$H = \text{Encoder}(T, \text{mask}) \tag{1.16}$$

where H represents the last_hidden_state. Unlike traditional word embeddings (e.g., Word2Vec), this model captures contextualized semantics, meaning the vector for a term like "Regulation" will vary depending on its surrounding legal clauses.

Though not ranking as high as Bertbased architecture in Benchmark, the significant smaller output dimension (only half of Bert with 768), is suitable for our empirical work. With expected output of more than 4000 labels, this backbone will avoid computational overhead, given our computation resources. By leveraging this pre-trained knowledge, the classification heads can focus on mapping pretrained model semantic output to EuroVoc concepts rather than learning from scratch.

1.4.2.2 Classification Head: BiLSTM-Attention

For the LSTM-based approach, we propose a Hybrid BiLSTM-Attention network. This model is designed to capture intersequential dependencies, thanks to BiLSTM network, while still highlight

critical legal "anchors", acting as key features within a that sequence.

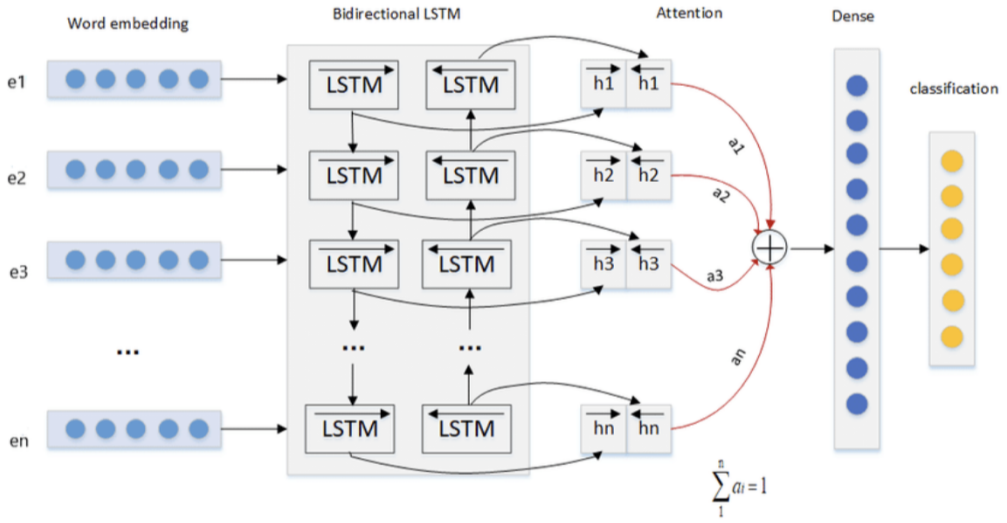


Figure 1.7: General architecture of BiLSTM with Self Attention

Bi-directional LSTM (BiLSTM) Layer

To capture complex dependencies in legal prose, the BiLSTM processes the input sequence X in both forward and backward directions:

$$\overrightarrow{h}_t = \text{LSTM}_{fwd}(x_t, \overrightarrow{h}_{t-1}) \tag{1.17}$$

$$\overleftarrow{h}_t = \text{LSTM}_{bwd}(x_t, \overleftarrow{h}_{t+1}) \tag{1.18}$$

$$h_t = [\overrightarrow{h}_t; \overleftarrow{h}_t] \tag{1.19}$$

The resulting hidden state $h_t \in \mathbb{R}^{2 \times d_{hidden}}$ provides a holistic representation of each token by considering both preceding citations and subsequent legal clauses.

Self-Attention Mechanism

Despite the improvements of LSTMs over standard RNNs, gradients can still vanish over long legal documents (3,000+ tokens) and deep layers. We implement a Self-Attention layer to compute a context vector c as a weighted sum of hidden states:

$$u_t = \tanh(W_a h_t + b_a) \tag{1.20}$$

$$\alpha_t = \frac{\exp(u_t^T u_w)}{\sum_i \exp(u_i^T u_w)}, \quad c = \sum_t \alpha_t h_t \tag{1.21}$$

This allows the model to "attend" to non-contiguous legal concepts, effectively mitigating information loss in long sequences.

1.4.2.3 Classification Head: Transformer-based

In our Architecture, the transformer acts as a custom multilayer attention heads to handle extreme multi-label classification, mapping output from pretrained model to corresponding EuroVoc labels.

Multi-segment Feature Fusion

To bypass the 512-token limit of the underlying backbone, we independently encode multiple segments (Title, Recitals, Body) and join them along the sequence dimension:

$$\mathbf{H}_{total} = \text{Concat}(H_{Title+Body}, H_{Recitals}, \text{dim} = 1) \in \mathbb{R}^{B\times 1024\times 384} \tag{1.22}$$

Internal Matrix Structure: The resulting matrix provides a continuous landscape for the classification head:

$$\mathbf{H}_{total} = \left[\begin{array}{c}\text{Segment 1: Title + Body Features } (512 \times 384)\\ \hline \text{Segment 2: Recital Features } (512 \times 384)\end{array}\right] \tag{1.23}$$

Flexible Attention Mechanisms

We implement two primary strategies for label prediction:

- Label-Wise Attention (LWA): Each label learns a unique query vector Q to "search" for relevant keywords within the fused 1024×384 feature space.
- Global Attention: Identifies the most significant information across the entire document to generate a global summary vector for classification.

Semantic Label Initialization (Warm-start)

To improve performance on rare labels, we "prime" the model by vectorizing the text of the labels themselves (e.g., "State aid"). The label is projected into a vector $V_{out} \in \mathbb{R}^{384}$. The weights of the final linear layer are initialized with these semantic vectors. Combined with a negative bias ($b = -2.0$), this ensures stable training and better recognition of Few-shot labels.

1.4.2.4 Optimization Strategy

Hybrid ASLCB Loss Function

To address extreme label imbalance, we combine Asymmetric Loss (ASL) with Class-Balanced (CB) weighting.

Class-Balanced Weights: Using the effective number of samples theory with $\beta = 0.999$, we assign weights W_i to each class. This results in a significant contrast to favor the Long-tail:

- Min Weight (Head): 1.0000 | Max Weight (Tail): 973.9262

By applying this, we balance the Information Gain (IG) across the dataset. A "Head" label (e.g., 1501 samples) has high redundancy, while a "Tail" label (e.g., 1 sample) carries 100% unique information. The 974× weight ensures rare signals are not washed out by frequent ones.

Asymmetric Loss with CB Integration: The final loss modulates Binary Cross-Entropy (BCE) using asymmetric focusing and clipping:

$$\mathcal{L} = -\frac{1}{N}\sum[y \cdot \alpha_{pos}\log(x_{pos}) + (1-y)\cdot\alpha_{neg}\log(x_{neg,\text{clipped}})] \times W_{CB} \tag{1.24}$$

Where $\gamma_{neg} = 2$ and $\gamma_{pos} = 0$ prioritize difficult negative samples, while W_{CB} amplifies the gradient of "Tail" labels by nearly 974×. With those loss Configuration, we have an empirical estimate of the information contribution and error distribution:

- Information Gain (IG) Comparison: The information content of a label can be estimated by its self-information $I(L) = -\log_2(P(L))$.
  - For a Head Label ($n = 1501$): $P(L_h) \approx 0.033$, resulting in $I(L_h) \approx 4.9$ bits.
  - For a Tail Label ($n = 1$): $P(L_t) \approx 0.000022$, resulting in $I(L_t) \approx 15.4$ bits.

  The Tail label provides over 3.1 times more surprise (information) per instance in a batch of 64, justifying the 974× weight to ensure its rare signal is captured.
- Negative Error Reduction: In a standard multi-label BCE, the negative samples overwhelm the positive ones. For a single document in Eurlex57k:
  - Positive Labels: Average 6 ($\approx 0.14\%$).
  - Negative Labels: Average 4,102 ($\approx 99.86\%$).

  By setting $\gamma_{neg} = 2$, we reduce the gradient contribution of "easy" negatives (where $x_{neg} < 0.05$) by a factor of over 400 times compared to standard BCE (1^2 vs 0.05^2). This effectively "cleans" the gradient, allowing the model to focus on the 0.14% of relevant legal signals.

Weight Optimizer

We employ the AdamW optimizer with a weight decay of 0.01 to provide decoupled weight regularization. The learning rate is managed using a Linear Warmup Scheduler, allowing the model (especially the Semantic-initialized heads) to stabilize before reaching the maximum learning rate.

1.4.3 Dataset Preparation

The dataset preparation phase bridges the gap between cleaned text and the numerical inputs required by each architecture. We utilize Multi-Label Binarization to transform the EuroVoc concepts into a multi-hot encoded vector $\mathbf{y} \in \{0, 1\}^{4108}$. To ensure consistency between training and inference, the `MultiLabelBinarizer` is fitted on the global set of filtered tags and persisted as a serialized object.

1.4.3.1 BiLSTM Dataset (Fixed Embedding Approach)

For the BiLSTM architecture, we prioritize computational speed by utilizing pre-computed embeddings. Instead of processing raw text during training, we represent each document as a single dense vector extracted from the `all-MiniLM-L12-v2` model.

1.4.3.2 Transformer Dataset (Multi-Segment Tokenization)

The Transformer dataset is designed to maximize context retention by splitting legal documents into two functional chunks, bypassing the standard 512-token truncation limit.

- Dual-Chunk Tokenization: We employ a `split_and_tokenize` strategy. Chunk 1 fuses the document title with the *Main Body*, while Chunk 2 isolates the *Recitals*. This ensures that the dense legal reasoning often found in the recitals is not discarded.
- Dynamic Encoding: Unlike the LSTM approach, this dataset provides raw `input_ids` and `attention_mask` tensors. This allows the model to learn fine-grained attention weights across a 1024-token context window ($512 + 512$).

```python
def split_and_tokenize(df, tokenizer, max_len=512):
    # Chunk 1: Title + Main Body
    c1 = f"{row['title']} [SEP] {row['main_body']}"
    # Chunk 2: Recitals (Key legal points)
    c2 = f"{row['recitals']}"
    enc1 = tokenizer(c1, padding='max_length', truncation=True, max_length=max_len)
    enc2 = tokenizer(c2, padding='max_length', truncation=True, max_length=max_len)
    return enc1, enc2
```

Listing 1.1: Transformer Multi-Segment Split

Label Mapping: To facilitate semantic initialization, we map concept IDs to their descriptive titles (e.g., "ID 889" $\rightarrow$ "State aid") using a mapping dictionary. This dictionary allows the Transformer to encode the labels themselves into the shared 384-dimensional space during the "Warm-start" phase.

1.5 Experiments: LSTM

This section details the experimental configuration, training progression, and performance evaluation of the BiLSTM-Attention architecture. The objective is to establish a baseline for sequential modeling on the Eurlex57k dataset.

1.5.1 Training

The BiLSTM model was trained using the AdamW optimizer with a learning rate of 10^{-3} and a batch size of 128. To handle the high-dimensional label space efficiently, we utilized Mixed Precision Training via PyTorch's `GradScaler`, significantly reducing memory overhead without compromising numerical stability.

Evaluation Metrics: In the context of extreme multi-label classification (XMC), we prioritize Macro-F1 to evaluate performance across the long-tail distribution. The metric is defined as the arithmetic mean of F1-scores calculated independently for each class $c \in C$:

$$\text{Macro-F1} = \frac{1}{|C|} \sum_{c \in C} \frac{2 \cdot P_c \cdot R_c}{P_c + R_c} \tag{1.25}$$

Additionally, we monitor Hamming Loss to assess the fraction of incorrectly predicted labels (both false positives and false negatives) relative to the total number of labels.

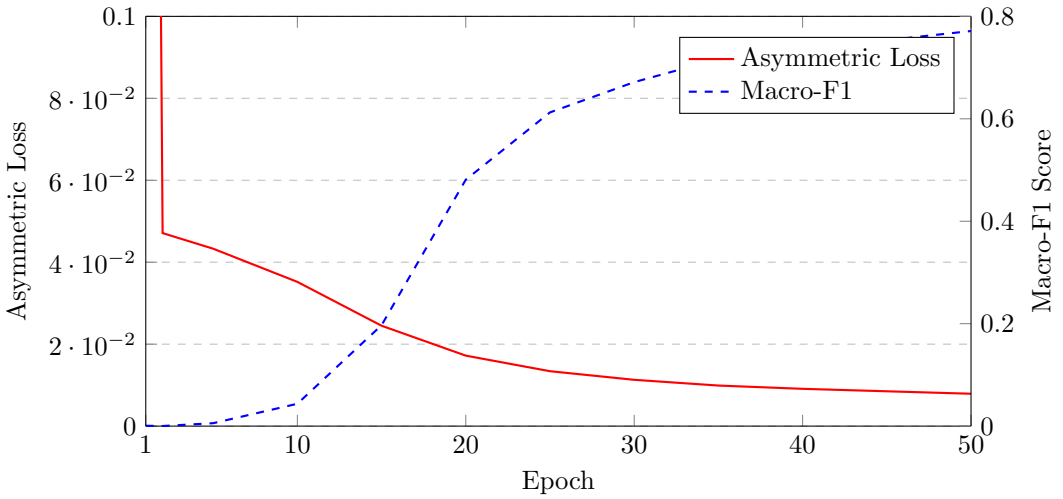


Figure 1.8: Training progression: Loss and Macro-F1 convergence.

The Macro F1 rise from epoch 20 onwards, proving that model understand features of all labels in dataset after that.

1.5.2 Evaluation and Results

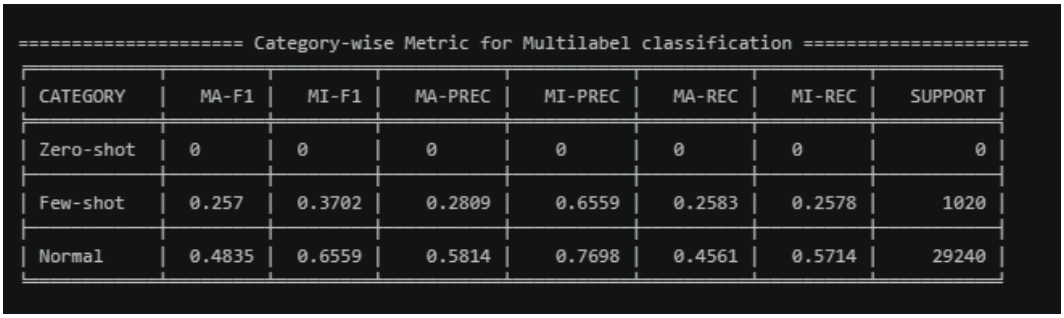
We conducted a comparative analysis of three LSTM variants: the base model (LSTM-Base), the model with Class-Balanced weights (LSTM-CB), and the model utilizing both Asymmetric Loss and CB weights (LSTM-ASLCB).

Table 1.6: Performance Comparison of LSTM Variants

Metric	LSTM-Base	LSTM-CB	LSTM-ASLCB
Micro F1	0.5731	0.6129	**0.6449**
Macro F1	0.2488	0.2580	**0.2469**
Precision@1	0.8577	0.8535	**0.8893**
NDCG@5	0.7445	0.7359	**0.7553**
Hamming Loss	0.00145	0.00084	**0.00076**

Detailed Category Analysis (ASLCB): The LSTM-ASLCB variant achieved the best overall precision and lowest Hamming Loss. However, a categorical breakdown reveals the inherent difficulty of the dataset:

- Normal Samples: Achieved a robust Micro-F1 of 0.6559.
- Few-shot Samples: Maintained a Micro-F1 of 0.3702, showing the effectiveness of the CB-weighting in recovering rare signals.
- Zero-shot Samples: As expected for a non-semantic LSTM architecture, the model failed to predict labels with zero training instances (F1 = 0).



==== Category-wise Metric for Multilabel classification ====

CATEGORY	MA-F1	MI-F1	MA-PREC	MI-PREC	MA-REC	MI-REC	SUPPORT
Zero-shot	0	0	0	0	0	0	0
Few-shot	0.257	0.3702	0.2809	0.6559	0.2583	0.2578	1020
Normal	0.4835	0.6559	0.5814	0.7698	0.4561	0.5714	29240

Figure 1.9: Catergory based performance

The results indicate that while the BiLSTM with Attention is highly precise for top-ranked predictions ($P@1 = 88.93\%$), it remains limited by its inability to handle semantic zero-shot generalization.

1.6 Experiments: Transformer

This section details the experimental configuration, training progression, and performance evaluation of the Transformer-based architecture (*Trans-ASLCB*). The objective is to evaluate the impact of context-aware embeddings and semantic initialization on the Eurlex57k extreme multi-label task.

1.6.1 Training

The Transformer model utilizes a tiered training strategy to balance the preservation of pre-trained knowledge with the requirement for domain-specific adaptation. The key configurations are as follows:

- Two-stage Fine-tuning: The training is divided into a Warm-up phase (31 epochs) and a Fine-tuning phase (9 epochs). During warm-up, the backbone (*all-MiniLM-L12-v2*) is frozen, allowing the *Label-Wise Attention* queries to stabilize.
- Differential Learning Rates: We apply a higher learning rate (10^{-3}) to the classification heads and a significantly lower rate (10^{-6}) to the backbone during the second phase to prevent catastrophic forgetting.
- Gradient Accumulation: To maintain a stable gradient signal given the high-dimensional label space, we set `acc_steps=4`, resulting in an effective batch size of 64.
- Loss Function (ASLCB): We employ the Asymmetric Loss combined with Class-Balanced weights ($\beta = 0.996$). Parameters are set to $\gamma_{neg} = 2.0$, $\gamma_{pos} = 0.5$, and a clipping value of 0.05 to suppress easy negative samples.

```
# Phase-based freezing and LR scheduling
if epoch < warm_up:
    model.transformer.requires_grad_(False)
    lr_backbone, lr_head = 0.0, 1e-3
else:
    model.transformer.requires_grad_(True)
```

```
    lr_backbone, lr_head = 1e-6, 5e-4
```

Listing 1.2: Transformer Training Logic

The convergence of the Trans-ASLCB model is visualized in Figure 1.10. A notable performance "jump" is observed at Epoch 31, coinciding with the unfreezing of the Transformer backbone.

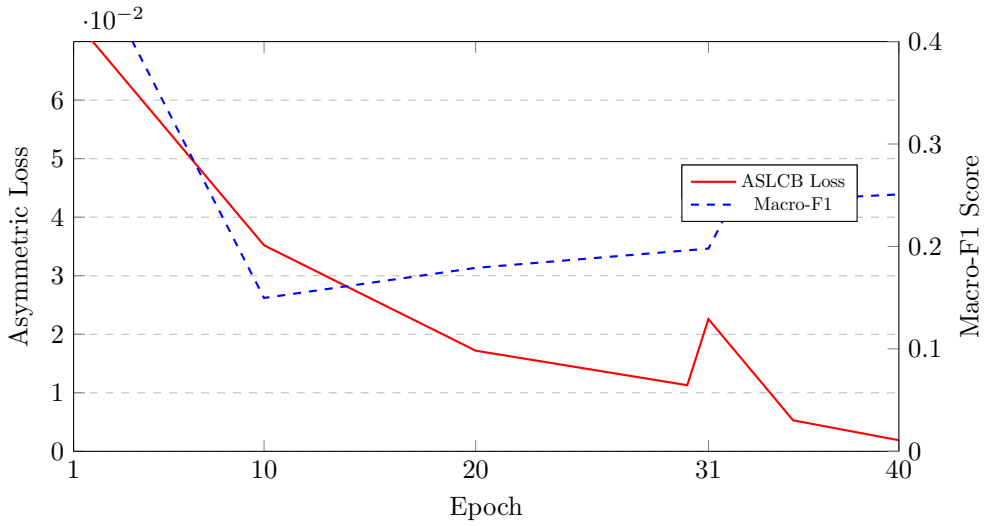


Figure 1.10: Training progression of Trans-ASLCB. The inflection point at Epoch 31 represents the start of the full-model fine-tuning phase.

1.6.2 Evaluation and Results

The best performance was recorded at Epoch 40, demonstrating the model's ability to maintain high precision even across a massive label set.

Table 1.7: Global Metrics for Trans-ASLCB at Best Epoch

Metric	Precision	Recall	F1-Score	Support
Micro Avg	0.5801	0.7122	0.6394	30,356
Macro Avg	0.2539	0.2828	0.2508	30,356
Weighted Avg	0.6070	0.7122	0.6347	30,356
Samples Avg	0.6068	0.7319	0.6396	30,356

1.6.3 Qualitative Analysis: Attention Rollout and Interpretability

To gain deeper insights into the decision-making process of the *Trans-ASLCB* model, we visualize the attention weights through an **Attention Rollout** analysis. This technique allows us to identify which specific tokens within a legal document contribute most significantly to the final multi-label prediction.

As illustrated in the attention heatmap (Figure 1.11), the model demonstrates a sophisticated ability to filter out structural noise and focus on high-entropy entities:

- Filtering Common Legal Jargon: Interestingly, the model assigns relatively low attention weights to frequent but non-discriminative legal terms such as *"amendment"*, *"article"*, or *"implementing"*. This suggests that the ASLCB Loss has successfully taught the model that these ubiquitous terms provide little information gain for specific EuroVoc classification.
- **Focus on Named Entities and Specific Context:** The attention mechanism exhibits high activation on specific entities and geographic markers, such as *"Singapore"* and "Commission". In the context of Eurlex57k, identifying the specific jurisdiction or the governing body is often a decisive factor for accurate labeling.
- **Segment Interaction:** The heatmap reveals strong attention bridges between Segment A (Title + Body) and Segment B (Recitals), particularly around tokens that define the scope of the legal act (e.g., "third countries", "approval").

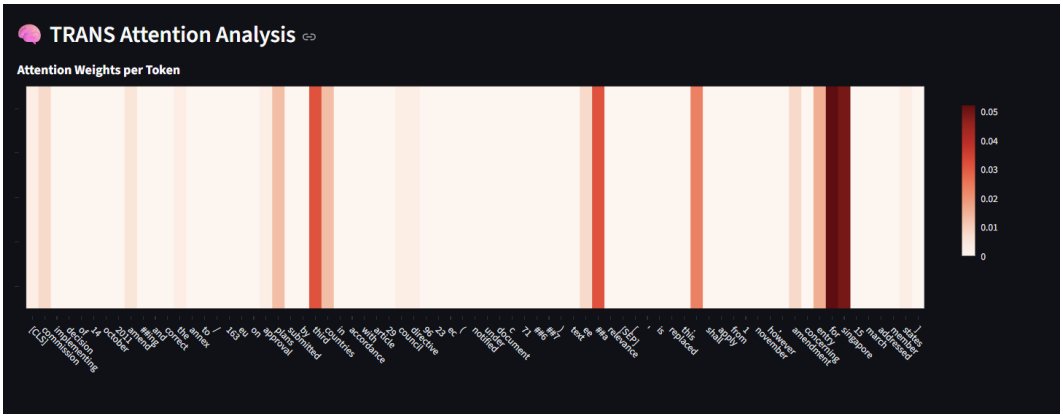


Figure 1.11: Attention Analysis: The model suppresses common functional words while focusing on high-impact entities like "Singapore" and "Commission" to derive semantic meaning.

The ability to ignore "filler" legal language in favor of specific nouns and entities confirms that our Transformer architecture, combined with **Semantic Warm-start**, effectively captures the core "legal essence" of the documents rather than merely performing keyword matching.

1.7 Final Evaluation

1.7.1 Comparison

The following table summarizes the performance of our proposed architectures alongside the official Eurlex57k State-of-the-Art (SOTA) benchmarks and our Vanilla MiniLM baseline.

Table 1.8: Consolidated Performance Benchmarking: SOTA, Baselines, and Proposed Models

Model	Micro F1	Macro F1	P@1	R@1	nDCG@5	Hamming
BERT-BASE * (SOTA)	0.6850	0.2200	0.9220	0.2100	0.8230	–
BIGRU-LWAN (L2V)	0.6120	0.1850	0.9130	0.1980	0.8040	–
HAN	0.5840	0.1420	0.8940	0.1820	0.7780	–
Vanilla MiniLM (Base)	0.0511	0.0314	0.1695	0.0347	0.1050	0.00160
BiLSTM-ASLCB (Ours)	0.6449	0.2469	0.8893	0.2026	0.7553	0.00076
Trans-ASLCB (Ours)	0.6394	0.2508	0.8810	0.2002	0.7608	0.00085

The experimental results demonstrate a massive performance increase when comparing our optimized architectures to the Vanilla MiniLM-L12-v2 baseline. The base model failed significantly in the extreme multi-label space, recording a Precision@1 of only 0.1695. By integrating Asymmetric Loss with Class-Balanced weights (ASLCB), we observed an improvement of over 70% in P@1 and increase in nDCG@5. This confirms that the vanilla encoder alone cannot overcome the gradient dominance of negative samples and that our specialized loss configuration is essential for learning rare legal concepts. Furthermore, our models achieve a higher Macro-F1 (0.250) than the BERT-BASE SOTA (0.220), indicating that our specialized classification head effectively addresses the long-tail distribution problem.

The comparison between the BiLSTM and Transformer architectures reveals distinct trade-offs. BiLSTM-ASLCB achieved the highest overall Micro F1 (0.6449) and the lowest Hamming Loss (0.00076), suggesting exceptional precision in suppressing false positives for fixed-length document embeddings. However, Trans-ASLCB yielded a superior Macro F1 (0.2508) and nDCG@5 (0.7608). Guided by Semantic Initialization, the Transformer demonstrated better capabilities in ranking labels by their actual legal relevance, approaching the performance of the hierarchical HAN model. While BiLSTM processes a single 384-dimensional vector, the Transformer's multi-segment fusion (1024 tokens) allows it to capture dependencies across a wider context window, driving improved recall across the long-tail EuroVoc hierarchy.

1.7.2 Application

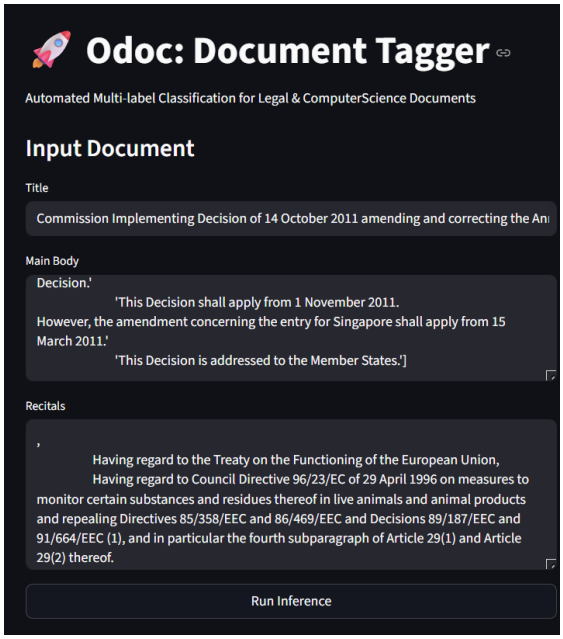
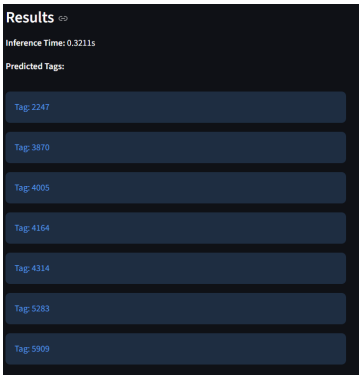
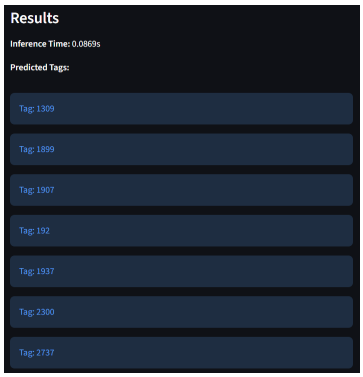


Figure 1.12: Applcation Interface

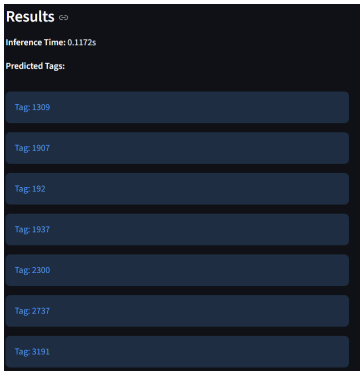
To bring the model into application we design an Inference PipeLine as demo, using Streamlit as GUI library. In order to fully understand the Law document, users input optionally 3 fields of Title, Body and Recitals. The result with inference time will be shown on the right. The inference outputs from our primary architectures are visualized in Figure 1.13. By comparing the three approaches, we observe how the ensemble strategy stabilizes the predictions.



(a) BiLSTM Prediction



(b) Transformer Prediction



(c) Ensemble ($w_{trans} = 0.7$)

Figure 1.13: Comparative Inference Results: (a) Sequential modeling output, (b) Attention-based semantic output, and (c) Final ensemble result optimized for high-precision legal indexing.

For Transformer, we provide visualization using Attention Rollout. Bassically the gradient

of the final layer will be taken and plotted with mapped token to show significant token within provided texts.

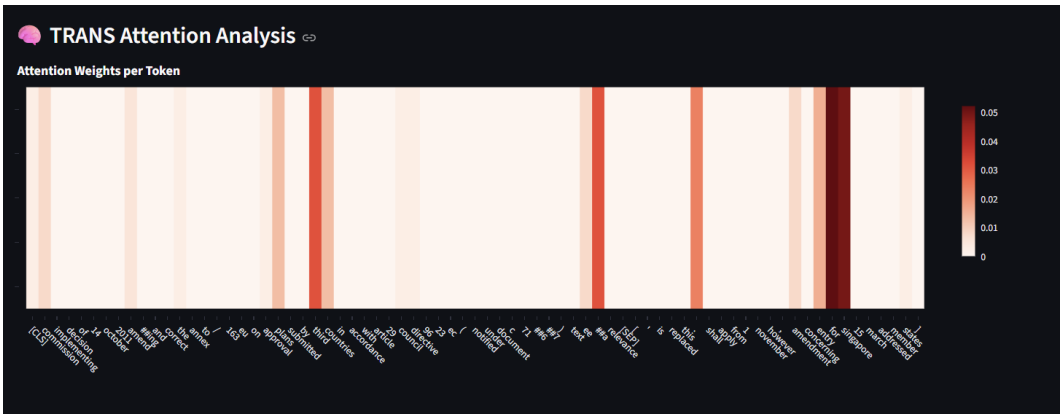


Figure 1.14: Attention Roll_out. The redder the colour the more focused the model on that token

1.8 Conclusion

Bibliography

[1] K. Bhatia, K. Dahiya, H. Jain, P. Kar, A. Mittal, Y. Prabhu, and M. Varma. The extreme classification repository: Multi-label datasets and code, 2016.

[2] Rohit Babbar and Bernhard Schölkopf. Data scarcity, robustness and extreme multi-label classification. Machine Learning and European Conference on Machine Learning, 2019.

[3] Yashoteja Prabhu, Anil Kag, Shrutendra Gopinath, Kunal Dahiya, Shrutendra Harsola, Rahul Agrawal, and Manik Varma. Extreme multi-label learning with label features for warm-start tagging, ranking and recommendation. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM), 2018.

[4] Sujay Khandagale, Han Xiao, and Rohit Babbar. Bonsai - diverse and shallow trees for extreme multi-label classification. arXiv preprint arXiv:1904.08249, 2019.

[5] Himanshu Jain, Venkatesh Balasubramanian, Bhanu Chunduri, and Manik Varma. Slice: Scalable linear extreme classifiers trained on 100 million labels for related searches. In Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM), 2019.

[6] Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. Extreme multi-label legal text classification: A case study in eu legislation. In Natural Legal Language Processing Workshop, 2019.

[7] Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. Large-scale multi-label text classification on eu legislation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), 2019.

[8] Ronghui You, Suyang Dai, Zihan Zhang, Hiroshi Mamitsuka, and Shanfeng Zhu. Attentionxml: Extreme multi-label text classification with multi-label attention based recurrent neural network. In Advances in Neural Information Processing Systems (NeurIPS), 2019.

[9] Siddhant Kharbanda, Ananye Banerjee, Deepti Gupta, Ashish Palrecha, and Rohit Babbar. Inceptionxml: A lightweight framework with synchronized negative sampling for short text extreme classification. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023.

[10] Han Ye, Rajshekhar Sunderraman, and Shihao Ji. Matchxml: An efficient text-label matching framework for extreme multi-label text classification. IEEE Transactions on Knowledge and Data Engineering (TKDE), 2024.

[11] Siddhant Kharbanda, Ananye Banerjee, R. Schultheis, and Rohit Babbar. Cascadexml: Rethinking transformers for end-to-end multi-resolution training in extreme multi-label classification. In Advances in Neural Information Processing Systems (NeurIPS), 2022.

[12] Anshul Mittal, Noveen Sachdeva, Sheetal Agrawal, Sumeet Agarwal, Purushottam Kar, and Manik Varma. Eclare: Extreme classification with label graph correlations. In Proceedings of the Web Conference (TheWebConf), 2021.

[13] Deepak Saini, Amit Kumar Jain, Kushal Dave, Jian Jiao, Amit Singh, Ruofei Zhang, and Manik Varma. Galaxc: Graph neural networks with labelwise attention for extreme classification. In Proceedings of the Web Conference (TheWebConf), 2021.