

Article

Hierarchical Classification of Arabic Legal Cases Using Transformer Architectures and Large Language Models

Nourah Alangari ^{1,*}, Nouf Alshenaifi ¹ and Huda Almuzaini ²¹ Computer Science Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia; noalshenaifi@ksu.edu.sa² Computer Science Department, College of Computer and Information Sciences, Imam Mohammed Ibn Saud Islamic University, Riyadh 11432, Saudi Arabia; hamozeani@imamu.edu.sa

* Correspondence: nmalangari@ksu.edu.sa

Abstract

Automated classification of Arabic legal texts presents unique challenges stemming from the formal register of judicial language, domain-specific Shari'a terminology, and the severe class imbalance inherent in hierarchical legal taxonomies. This paper addresses these challenges through a systematic investigation of hierarchical multi-class classification applied to a dataset of 1146 Arabic judicial cases sourced from the Saudi Ministry of Justice open data portal. Cases are annotated at two hierarchical levels: a binary main-class distinction between criminal and civil cases (L_1), and a fine-grained sub-category classification across 63 legal topics (L_2), exhibiting a class imbalance ratio of 43:1. We evaluate three methodological families under four classification architectures. For encoder-only models, we fine-tune AraBERT, CAMEL-BERT, MARBERTv2, and XLM-RoBERTa under Flat (Bottom-Up), Hierarchical Multi-Task, Two-Stage, and Hierarchy-Aware configurations. For encoder-decoder models, we adapt AraT5v2 under the same four architectures. Finally, we conduct a prompt-based evaluation of GPT-4o under zero-shot and targeted few-shot settings. The experimental results demonstrate that the Hierarchy-Aware architecture consistently achieves the strongest fine-grained classification performance across both model families. AraBERTv2 with Hierarchy-Aware training achieves the best overall L_2 accuracy of 83.04% and a Macro-F1 of 76.07%, while the Single Multi-Task configuration achieves the highest L_1 accuracy of 99.57%. GPT-4o under 5-shot prompting achieves 99.13% L_1 accuracy and a competitive L_2 Macro-F1 of 74.30% without task-specific fine-tuning, though supervised models maintain stronger overall fine-grained performance. These findings highlight the importance of explicit hierarchical supervision and domain-adapted pre-training for Arabic legal text classification, and establish strong baselines for future research in this underexplored domain.



Academic Editor: Janos Botzheim

Received: 5 June 2026

Revised: 3 July 2026

Accepted: 5 July 2026

Published: 8 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: Arabic legal text classification; hierarchical classification; Arabic NLP; transformer models; AraBERT; AraT5; GPT-4o; large language models; legal AI

1. Introduction

The rapid expansion of digitized legal records across the Arab world has created a pressing need for automated tools capable of organizing, navigating, and analyzing vast corpora of judicial documents. Courts in the Kingdom of Saudi Arabia alone produce thousands of case judgments annually, each grounded in a complex interplay of civil statutes and Shari'a-based legal reasoning. Manually categorizing such documents is both time-

consuming and error-prone, motivating the development of robust automated classification systems that can operate on formal Arabic legal text with high accuracy and structural consistency.

Legal text classification is an active research area, with transformer-based models [1] having substantially advanced the state of the art across English and European legal corpora [2]. However, Arabic legal text remains critically underrepresented in the natural language processing literature. The linguistic characteristics of formal Arabic—including complex morphology, diacritics, and domain-specific Shari’a terminology—render direct transfer from English or multilingual models non-trivial. Moreover, Arabic legal taxonomies often exhibit pronounced hierarchical structure, where each case must be assigned both a coarse-grained main category and a fine-grained sub-category, introducing additional challenges of label co-occurrence, class imbalance, and hierarchical consistency that have not been adequately addressed in prior work.

This paper makes the following contributions:

- (1) Systematic empirical comparison in an Arabic legal context. We evaluate two families of fine-tuned transformer models—four encoder-only models (AraBERT, CAMeL-BERT, MARBERTv2, and XLM-RoBERTa) and one encoder-decoder model (AraT5v2)—alongside a prompt-based evaluation of GPT-4o on hierarchical Arabic legal case classification.
- (2) Comparative evaluation of hierarchical classification strategies. We compare four existing classification strategies: Flat (Bottom-Up), Hierarchical Multi-Task, Two-Stage Pipeline, and Hierarchy-Aware classification with a structural consistency penalty. The goal is to enable a fine-grained analysis of how architectural choices affect performance at each level of the hierarchy in the context of Arabic legal case classification.
- (3) Benchmark results for an underexplored domain. The study provides empirical baselines and analysis for Saudi Arabic legal case classification, a domain that remains underrepresented in Arabic NLP research.

The remainder of the paper is organized as follows. Section 2 reviews related work on legal text classification, hierarchical architectures, and Arabic NLP. Section 3 describes the dataset construction, annotation scheme, and statistical characteristics. Section 4 presents the proposed classification framework, including task definition, model descriptions, classification architectures, training setup, and evaluation metrics. Section 5 reports experimental results across all model families and architectures. Section 6 discusses the key findings, model comparisons, and practical implications. Section 7 concludes the paper and outlines directions for future work.

2. Related Work

Research on automated legal text classification has advanced considerably, progressing from traditional machine learning through deep learning to transformer-based and large language models. Early work relied on feature engineering: Chen et al. [3] showed that a domain concept-based Random Forest using TF-IDF features outperformed deep learning models (TextCNN, BiLSTM, BERT embeddings) on 30,000 U.S. case documents, achieving 85.98% accuracy. For explainability, González-Grandes et al. [4] combined jurisdiction-dependent Random Forest classifiers with a natural-language explanation module validated by legal experts at above 90% accuracy on 96,163 Spanish judgments. For shorter or simpler tasks, Tomar et al. [5] fine-tuned BERT-Base on just 393 cases for binary civil/criminal classification, achieving 92.9% accuracy.

Long document classification has motivated hierarchical and efficient architectures. Prasad et al. [6] demonstrated that stacking a fine-tuned LegalBERT chunk encoder with a two-layer Bi-RNN improved legal judgment prediction on ILDC by approximately 3 metric

points over prior baselines, confirming the value of domain-specific pre-training combined with hierarchical processing. Wu et al. [7] combined sliding-window local attention, clustering-based long-range attention, and global tokens into a single transformer, outperforming Longformer, Big Bird, and Fastformer on the IMDB and CAIL-Long datasets.

Multi-label legal classification introduces further challenges around label co-occurrence, class imbalance, and long-tail distributions. Song et al. [8] introduced POSTURE50K (50,000 U.S. legal opinions) and proposed LAMT_MLC, combining domain-specific BERT pre-training with a label-attention mechanism and multi-task learning for rare labels, outperforming baselines on both POSTURE50K and EUROLEX57K. Tuggener et al. [9] introduced LEDGAR, a large-scale multilabel corpus of SEC contract provisions containing hundreds of thousands of provisions with over 12,000 labels, designed to support research on legal text classification and provision analysis. Xu et al. [10] introduced the VECHR dataset for vulnerability classification in European Court of Human Rights cases, benchmarking BERT, LegalBERT, and hierarchical variants and finding significant out-of-domain performance drops. Papaloukas et al. [11] introduced the Greek Legal Code (GLC) dataset of 47,000 legislative resources, finding that monolingual GREEK-LEGAL-BERT achieves best results. Sargeant et al. [12] applied Claude 3 Opus with a self-evaluation prompting mechanism to 3078 UK summary judgments under a novel taxonomy, achieving 87.13% accuracy.

Transformer and encoder-decoder architectures have further pushed classification performance. Vatsal et al. [13] evaluated chunk-based, summarization-based, voting, and Stride-64 sliding-window strategies across BERT, RoBERTa, and LegalBERT on 8419 SCDB documents, with Stride-64 achieving 80% accuracy on 15 categories and 60% on 279 categories, improving state-of-the-art by 8% and 28%, respectively. Most directly relevant to our work, Kementchedjhieva et al. [14] compared four architectures—Encoder + Head, LWAN, Seq2Seq, and T5Enc (non-autoregressive T5 decoder)—on UKLEX, EURLEX, and BIOASQ at multiple label granularities, finding that encoder-decoder methods consistently outperform encoder-only methods. This finding directly motivates applying encoder-decoder models to Arabic legal text.

For large-scale legal annotation, Dominguez-Olmedo et al. [15] benchmarked commercial APIs (GPT-4.5, Claude 3.7) and open-source models (Llama 3, Qwen) alongside fine-tuned Lawma models on 260 classification tasks from 24,916 U.S. court opinions, showing that fine-tuned Lawma 8B achieves up to 87% accuracy—surpassing all commercial models with only a few hundred labeled examples. Weichbroth et al. [16] similarly evaluated ChatGPT on 268 Polish Penal Code cases across 15 crime types, achieving perfect binary classification, though with limited generalizability due to the small dataset and restricted crime types. For cross-lingual settings, Chalkidis et al. [2] introduced MULTI-EURLEX, 65,000 EU laws in 23 languages with four label granularities, and showed that adapter modules substantially improve zero-shot cross-lingual transfer over full fine-tuning for XLM-RoBERTa.

Several studies have addressed Arabic legal NLP more directly. Ikram and Chakir [17] investigated Arabic legal text classification using Moroccan legal documents, representing one of the earlier efforts in Arabic legal document classification. Al-Qurishi et al. [18] introduced AraLegal-BERT, a domain-specific Arabic legal language model, and showed the benefit of legal-domain pre-training for Arabic legal NLP tasks. More recently, Bouhouche et al. [19] classified Moroccan legal and legislative texts using machine learning models, while Ammar et al. [20] explored Arabic legal ruling prediction using large language models. Tahtah et al. [21] further studied Arabic legal question classification using domain-specific word embeddings and deep learning models. Despite these contributions, existing Arabic legal NLP studies mainly focus on legal language modeling, legal question classification, ruling prediction, or general legal document classification. Fine-grained hier-

archival classification of Arabic judicial cases under severe class imbalance remains comparatively underexplored, particularly in the context of Saudi legal texts and hierarchical legal taxonomies.

To bridge this gap, we investigate hierarchical classification of Arabic legal text using fine-tuned transformer-based models, including AraBERT and CAMEL-BERT. We further conduct an exploratory evaluation of large language models (LLMs) under zero-shot and few-shot prompting settings. This enables a comparative analysis between supervised and prompt-based approaches in a specialized and underrepresented legal context.

3. Dataset

We introduce a novel Arabic legal text dataset constructed from judicial case records obtained through the official open data portal of the Saudi Ministry of Justice (It is worth noting that these published volumes were previously accessible through the Ministry's official website; however, they are no longer publicly available at the time of writing. The dataset used in this study was collected prior to their removal.) The dataset is derived from publicly accessible judicial materials published as part of the Ministry's open data initiative. The source materials are organized into 28 published volumes in PDF format, each containing multiple judicial cases. Figure 1 shows an example of a single case extracted from Volume 1.

The annotation and classification of cases were performed by a specialized scientific team within the Ministry of Justice, comprising qualified Sharia consultants and legal researchers, supported by a delegated judge. This team is responsible for sorting and classifying the rulings according to a standardized taxonomy approved by the Permanent Scientific Committee, preparing case summaries that capture the judge's decision and the legal reasoning, and anonymizing the documents by removing identifying information. We extracted each case from the PDF files and assigned the hierarchical category labels (Main Class and Sub Class) based on the sectional organization within the published volumes.



Figure 1. An example of a judicial case extracted from Volume 1 of the dataset. The section headings are shown in Arabic (e.g., Subject, Legal Basis, and Case Summary).

The collected data consists of authentic court judgments adjudicated across general courts in the Kingdom of Saudi Arabia. Each record is preserved in its original Arabic form

without linguistic modification, maintaining the formal legal structure and terminology of the source documents. The dataset is structured into five key fields: case number, topics, legal basis, case summary, and full judgment text. Records with incomplete or missing information were removed to ensure data quality and consistency.

After preprocessing and filtering low-frequency categories, the final dataset comprises 1146 judicial cases. Each case is annotated using a hierarchical classification scheme consisting of two levels: (1) a Main Class (L_1) distinguishing between criminal and civil cases, and (2) a Sub Class (L_2) providing fine-grained categorization across 63 legal topics.

The distribution of the main classes is moderately balanced, consisting of 644 criminal cases and 502 civil cases. However, at the sub-class level, the dataset exhibits a pronounced long-tail distribution. The most frequent class contains 131 instances, while the least frequent classes contain only 3 instances, resulting in a severe imbalance ratio of approximately 43:1. Furthermore, 20 sub-classes contain fewer than 5 samples, and 32 sub-classes contain fewer than 10 samples, highlighting the difficulty of fine-grained classification in this domain. As shown in Figure 2, the main class distribution is moderately balanced. However, Figure 3 illustrates the pronounced long-tail distribution at the sub-class level, while Table 1 summarizes the dataset statistics.



Figure 2. Distribution of main classes (L_1) in the dataset. The figure uses the original Arabic class labels: “جناي” (Criminal) and “حقوقي” (Civil). The dataset is moderately balanced, with 644 criminal cases and 502 civil cases.

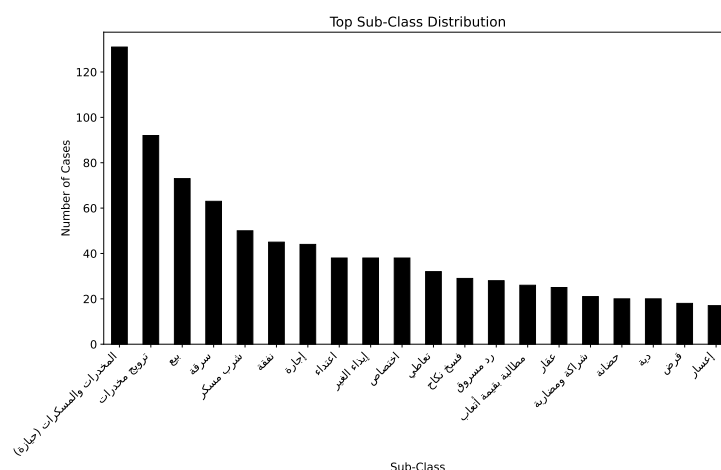


Figure 3. Top sub-class (L_2) distribution in the dataset. The figure highlights the long-tail nature of the data, where a small number of sub-classes dominate while many others have relatively few samples. Note: For readability, the Arabic sub-class labels shown on the x-axis correspond, from left to right, to: (1) Narcotics Possession, (2) Drug Trafficking, (3) Sale, (4) Theft, (5) Drinking Intoxicants, (6) Alimony/Maintenance, (7) Lease/Tenancy, (8) Assault, (9) Harming Others, (10) Jurisdiction, (11) Drug Consumption, (12) Annulment of Marriage, (13) Return of Stolen Property, (14) Claim for Professional Fees, (15) Real Estate, (16) Partnership and Speculation, (17) Blood Money (Diyya), (18) Child Custody, (19) Loan, and (20) Insolvency.

In terms of textual characteristics, the case summaries have an average length of 191.6 words (standard deviation: 94.7), with lengths ranging from 35 to 805 words. In contrast, the full judgment texts are substantially longer, with an average length of 1452.5 words (standard deviation: 932.1), and a maximum length of 6344 words. These statistics motivate the use of summary-based representations for efficient modeling, while retaining full judgments for future long-document analysis. For all experiments, the data was split into an 80/20 ratio for training and testing using stratified sampling by sub-class label, ensuring proportional representation of all 63 sub-classes in both sets.

Table 1. Summary statistics of the filtered Arabic legal dataset derived from the Saudi Ministry of Justice open data portal.

Statistic	Value	Details
Total cases	1146	Filtered dataset
L_1 : Criminal	644 (56.2%)	
L_1 : Civil	502 (43.8%)	
L_2 unique sub-classes	63	Long-tail distribution max = 131, min = 3
Imbalance ratio (L_2)	43:1	
Avg. summary length (words)	191.6 ± 94.7	
Avg. judgment length (words)	1452.5 ± 932.1	max = 6344
Classes < 5 samples	20	Rare classes
Classes < 10 samples	32	Highly sparse classes

4. Proposed Framework

4.1. Task Definition

Each lawsuit in the dataset consists of four sections: the subjects of the case, which represent the main topics addressed in the lawsuit; the legal basis of the ruling, which refers to the Sharia reasoning upon which the judgment was made; the case summary, which provides a concise overview of the facts and circumstances; and the text of the ruling, which contains the full details of the case across all court sessions.

For the purpose of classification, we selected only the case summary as the input text, as it captures the essential facts in a condensed form that is sufficient for category prediction while being computationally more efficient than processing the full ruling text.

All approaches target the same two-level hierarchical classification task:

- Level 1 (L_1): Main legal category classification (criminal vs. civil).
- Level 2 (L_2): Fine-grained sub-category classification across 63 legal classes.

Figure 4 illustrates the overall workflow of the proposed framework. The system takes the case summary as input and evaluates multiple pre-trained transformer models and prompt-based LLM settings under different hierarchical classification architectures to predict both the main legal category (L_1) and the fine-grained sub-category (L_2).

4.2. Pre-Trained Language Models

4.2.1. Encoder-Only Models

Four pre-trained encoder-based transformer models were evaluated as backbone classifiers:

- **AraBERT** (aubmindlab/bert-base-arabertv2) [22]: pre-trained on a large-scale Arabic corpus consisting of news articles, Wikipedia, and social media text, providing a strong general-purpose Arabic baseline.
- **CAMEL-BERT** (CAMEL-Lab/bert-base-arabic-camelbert-mix) [23]: trained on a diverse 167 GB Arabic corpus combining Modern Standard Arabic and dialectal content, making it well-suited to formal Arabic text.

- **MARBERTv2** (UBC-NLP/MARBERTv2) [24]: pre-trained on approximately one billion Arabic tweets (~128 GB), capturing both Modern Standard Arabic and diverse dialectal varieties.
- **XLM-RoBERTa** (xlm-roberta-base) [25]: a multilingual masked language model pre-trained on 2.5 TB of filtered CommonCrawl data across 100 languages, including Arabic. Although not specifically designed for Arabic, its multilingual representations provide a useful baseline for evaluating the importance of language-specific pre-training.

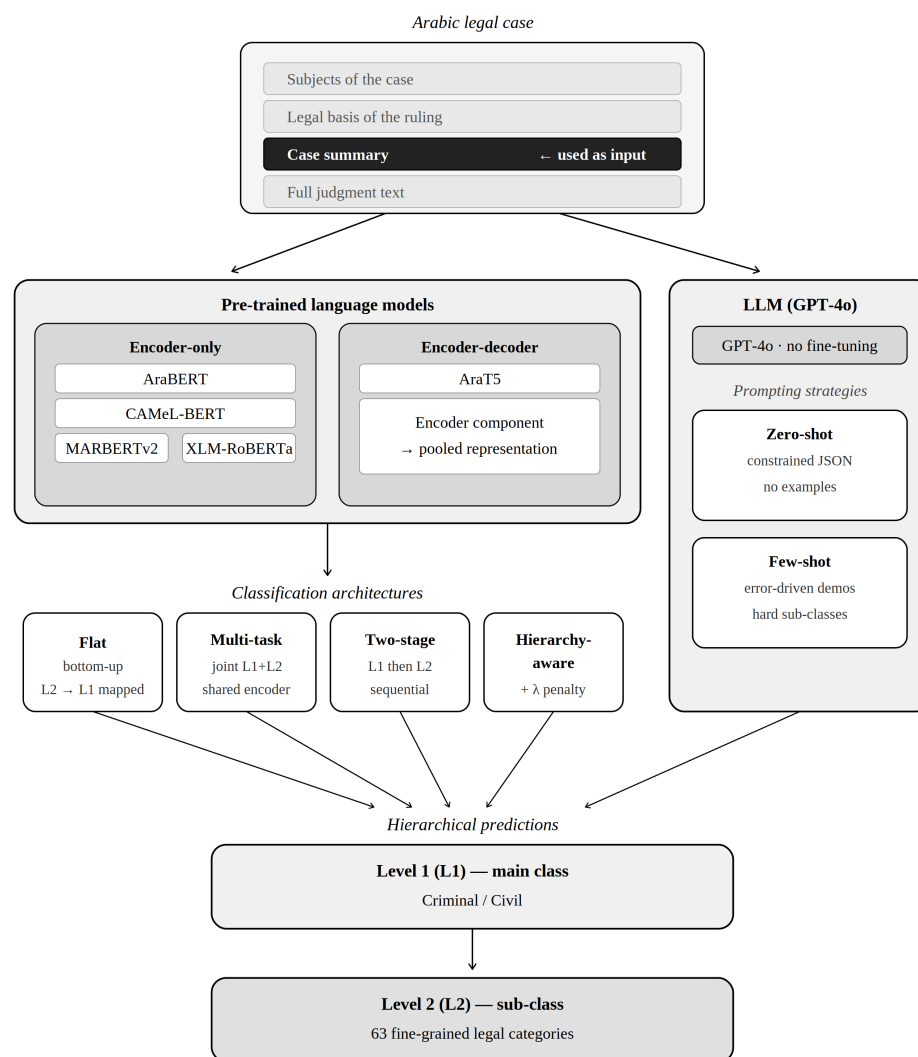


Figure 4. Overall framework of the proposed hierarchical Arabic legal classification system. The case summary is selected as input and evaluated using both pre-trained transformer models and GPT-4o under multiple classification architectures and prompting strategies to predict hierarchical legal categories. The arrows indicate the flow of data and processing from the input case summary through the classification models to the hierarchical predictions.

4.2.2. Encoder–Decoder Model

For the encoder–decoder family, we employ **AraT5v2** (UBC-NLP/AraT5v2-base-1024) [26], an Arabic sequence-to-sequence transformer model. The encoder component produces a pooled representation that is passed to downstream classification heads.

4.3. Classification Architectures

Each backbone model was evaluated under four classification architectures: (1) Flat (Bottom-Up) Classification, (2) Hierarchical Multi-Task Classification, (3) Two-Stage Hierarchical Pipeline, and (4) Hierarchy-Aware Classification.

4.3.1. Flat (Bottom-Up) Classification

The task is formulated as a single-label classification problem over the 63 fine-grained L_2 sub-classes. The encoder produces a pooled representation that is passed to a single classification head optimized using standard cross-entropy loss over L_2 labels only. Hierarchical relationships are not explicitly modeled during training. Instead, the corresponding L_1 category is recovered deterministically at inference time through a predefined mapping from each L_2 sub-class to its parent category. This configuration serves as a baseline for evaluating implicit hierarchical learning without explicit supervision.

4.3.2. Hierarchical Multi-Task Classification

This configuration employs a unified multi-task learning framework in which a shared encoder simultaneously predicts both L_1 and L_2 through two independent linear classification heads attached to the shared [CLS] token representation.

A dropout layer ($p = 0.1$) is applied before both heads for regularization. The model is optimized using a weighted combination of the main-class cross-entropy loss, $\mathcal{L}_{\text{main}}$, and the sub-class cross-entropy loss, $\mathcal{L}_{\text{sub}}$:

$$\mathcal{L} = 0.2 \cdot \mathcal{L}_{\text{main}} + 0.8 \cdot \mathcal{L}_{\text{sub}} \quad (1)$$

The larger weight assigned to the sub-category loss reflects the greater complexity of fine-grained classification relative to the binary main-category task. Joint optimization encourages the shared encoder to learn representations that benefit both classification levels simultaneously, and avoids the cascading-error problem of the sequential pipeline.

4.3.3. Two-Stage Hierarchical Pipeline

This approach explicitly mirrors the hierarchical taxonomy of legal case categories through a sequential two-stage pipeline.

In the first stage, the encoder is fine-tuned to predict the main category L_1 (criminal vs. civil). The predicted L_1 label is then used to route the input to a dedicated second-stage classifier trained exclusively on the corresponding subset of L_2 sub-classes.

A separate classifier is independently fine-tuned for each top-level branch. This decomposition reduces the effective label space and allows each second-stage model to specialize in semantically related sub-categories.

However, this architecture is susceptible to cascading errors, where an incorrect L_1 prediction inevitably leads to incorrect L_2 classification.

4.3.4. Hierarchy-Aware (L_1 -Conditioned) Classification

This configuration explicitly models the two-level hierarchy by jointly predicting both L_1 and L_2 through two separate classification heads attached to the shared encoder. To promote structural coherence, a hierarchy-consistency penalty is incorporated into the training objective:

$$\mathcal{L} = \mathcal{L}_{L_1} + \mathcal{L}_{L_2} + \lambda \cdot \mathcal{L}_{\text{hierarchy}} \quad (2)$$

The hierarchy penalty is computed by first constructing a binary validity mask $M \in \{0, 1\}^{|L_1| \times |L_2|}$ from the training taxonomy, where $M_{ij} = 1$ if sub-class j is a valid child of main class i , and $M_{ij} = 0$ otherwise. During training, the predicted L_2 proba-

bility distribution is obtained using a softmax function, and the hierarchy penalty is defined as the average probability mass assigned to sub-classes that are invalid under the ground-truth L_1 label. The regularization coefficient was fixed at $\lambda = 0.1$ for all hierarchy-aware experiments.

During inference, the same validity mask is used to suppress invalid parent–child combinations by assigning a large negative value to incompatible L_2 logits prior to prediction, thereby guaranteeing that every predicted sub-class remains consistent with the predicted parent category. This hard structural constraint makes the hierarchy-aware model particularly suited to operational legal classification systems, where taxonomically inconsistent outputs undermine interpretability and trustworthiness.

4.4. Large Language Model Evaluation

GPT-4o was evaluated under both zero-shot and few-shot prompting settings, representing a prompt-based paradigm that requires no gradient-based fine-tuning.

Both settings use the same held-out test set of 230 legal case summaries and representing 20% of the entire dataset. Performance was evaluated using:

- Main Class (L_1) Accuracy, Precision, Recall;
- Sub-Class (L_2) Accuracy, Precision, Recall;
- Exact Match Accuracy (both levels simultaneously correct);
- Hierarchy Compliance (whether the predicted L_2 sub-class is a valid child of the predicted L_1 class).

4.4.1. Zero-Shot Prompting

GPT-4o is provided with a legal case summary together with a constrained prompt defining the valid values for both Main_Class and Sub_Class drawn from the full taxonomy. The prompt explicitly instructs the model to select labels only from the predefined label set and to return its prediction in a structured JSON format. No task-specific examples are included. This setting evaluates the model’s inherent capacity for legal reasoning without any exposure to the target distribution.

4.4.2. Few-Shot Prompting

A targeted few-shot strategy is introduced to assess whether providing task-specific demonstrations improves classification performance. Rather than selecting examples randomly, demonstrations are drawn from the sub-classes that exhibited the highest error rates during zero-shot evaluation—specifically, semantically overlapping categories such as compensation-related criminal cases and related civil claims. The objective is to supply the model with representative decision boundaries for the categories most frequently confused, rather than broadly sampling the training distribution.

All few-shot demonstrations were selected exclusively from the original training partition and did not include any examples from the GPT-4o evaluation set.

All GPT-4o experiments were conducted using the OpenAI GPT-4o API (accessed in April 2026) under deterministic inference settings (temperature = 0.0, max_tokens = 80, and JSON-mode output). The same configuration was applied across the zero-shot, 3-shot, and 5-shot experiments. The prompt consisted of a system instruction defining the hierarchical legal taxonomy and classification constraints, followed by the legal case summary as user input. The complete prompts used in the GPT-4o experiments are provided in Appendix A. Since deterministic inference was used, a single run was performed for each case. Model outputs were validated using an automatic JSON parser; in the event of malformed or invalid responses, the API request would have been repeated. However, no invalid JSON outputs or API errors were observed in the final experiments.

4.5. Training Setup

All experiments were conducted using Google Colab GPU runtimes with an NVIDIA Tesla T4 GPU. For all fine-tuned transformer-based models, the input text consisted of the case summary only, rather than the full judgment text, in order to reduce computational cost and ensure consistency across models. The maximum sequence length was set to 512 tokens; longer summaries were truncated and shorter summaries were padded.

All fine-tuned models were trained using the AdamW optimiser with a weight decay of 0.01 and a batch size of 8. No gradient accumulation was used; the reported batch size refers to the actual per-device batch size used during training. The learning rate was selected from the search space $\{2, 2.5, 3, 3.5, 4, 4.5, 5\} \times 10^{-5}$, with 3×10^{-5} yielding the best validation performance across all configurations. A held-out test set was reserved exclusively for final evaluation and was never used during training or model selection. Early stopping based on validation loss was applied to prevent overfitting, with the best-performing checkpoint selected for evaluation. A dropout rate of 0.1 was applied.

- **Encoder-Only (Two-Stage):** The L_1 classifier was trained with 500 warm-up steps, while each L_2 classifier was trained independently with 10 warm-up steps.
- **Encoder-Only (Multi-Task):** Training was conducted with 50 warm-up steps.
- **AraT5 Models:** For hierarchy-aware training, the penalty coefficient was fixed at $\lambda = 0.1$.

4.6. Baseline Models

To provide a fair comparison and quantify the effectiveness of the proposed transformer-based models, two traditional machine learning baselines were implemented using TF-IDF text representations. These baseline models serve as reference methods to evaluate the performance gains achieved by the proposed deep learning approach.

The first baseline employs Term Frequency–Inverse Document Frequency (TF-IDF) features with a Logistic Regression (LR) classifier. Logistic Regression is a widely adopted linear classifier for text classification tasks due to its simplicity, computational efficiency, and strong performance on high-dimensional sparse feature spaces.

The second baseline combines TF-IDF features with a Linear Support Vector Machine (Linear SVM). Linear SVM has been extensively used as a strong benchmark for document classification because of its robustness and ability to effectively separate high-dimensional textual representations.

4.7. Evaluation Metrics

All models were evaluated using accuracy, precision, recall, and Macro-F1 at both hierarchical levels, namely L_1 and L_2 . Accuracy measures the overall proportion of correctly classified cases. Given the pronounced long-tail class imbalance at the sub-class level, Macro-F1 was used as the main F1-score because it gives equal weight to all classes regardless of their frequency. This is particularly important since many L_2 sub-classes contain very few samples. For LLM experiments, Exact Match Accuracy and Hierarchy Compliance are also reported.

5. Results

In this section, we present the experimental results of the proposed hierarchical Arabic legal text classification framework. The evaluation is organized according to the three methodological families introduced in the proposed framework section: (1) baseline models, (2) encoder-only transformer models, (3) encoder–decoder transformer models, and (4) prompt-based large language models. The experiments investigate the effectiveness of different hierarchical classification architectures, including Flat (Bottom-Up),

Hierarchical Multi-Task, Two-Stage Hierarchical Pipeline, and Hierarchy-Aware classification. Performance is evaluated at both hierarchical levels: the main legal category (L_1) and the fine-grained sub-category level (L_2), using accuracy, precision, recall, and F1-score. In addition, hierarchy compliance and exact match accuracy are reported for prompt-based LLM experiments.

5.1. Baseline Model Results

The experimental results of the baseline models are presented in Table 2. As expected, the majority-class baseline achieved substantially lower performance than the learning-based approaches, particularly for the fine-grained Sub-Class classification task, confirming the non-trivial nature of the classification problem. Among the learning-based baselines, Linear SVM consistently outperformed Logistic Regression for both classification tasks. For Main-Class classification, both traditional machine learning models achieved high performance, with Linear SVM obtaining a Macro-F1 score of 98.68%, indicating that TF-IDF features are effective for distinguishing broad legal categories.

Table 2. Performance of traditional machine learning baselines. Bold values indicate the best performance in each column.

Category Level	Model	Accuracy	Precision	Recall	Macro-F1
Main Class	Majority Class	0.5609	0.2804	0.5000	0.3593
	TF-IDF + Logistic Regression	0.9652	0.9633	0.9690	0.9650
	TF-IDF + Linear SVM	0.9870	0.9856	0.9884	0.9868
Sub-Class	Majority Class	0.1130	0.0019	0.0169	0.0034
	TF-IDF + Logistic Regression	0.5522	0.2304	0.2272	0.2120
	TF-IDF + Linear SVM	0.7565	0.5978	0.5599	0.5586

In contrast, Sub-Class classification was more challenging. Although Linear SVM achieved a substantially higher Macro-F1 score (55.86%) than Logistic Regression, its performance remained noticeably lower than that of the Main-Class task. This reflects the increased complexity of fine-grained legal category classification.

Overall, the baseline models provide a strong reference for evaluating the proposed transformer-based models. While TF-IDF combined with Linear SVM performs well for coarse-grained classification, its effectiveness decreases for fine-grained classification because it relies on sparse lexical features and cannot capture contextual semantic information. As shown in the subsequent experiments, the transformer-based models achieve superior performance by learning richer contextual representations of legal texts.

5.2. Encoder-Only Model Results

Tables 3 and 4 present the performance of the encoder-only architectures across the four hierarchical classification strategies: Single Multi-Task, Two-Stage, Bottom-Up (Flat), and Hierarchy-Aware. Performance is evaluated separately for the main-category classification task (Level 1) and the fine-grained sub-category classification task (Level 2).

For Level 1 classification, the Single Multi-Task configuration achieved the strongest overall performance among the encoder-only approaches. AraBERTv2-base obtained the highest Level 1 accuracy of 99.57% and a Macro F1-score of 0.9956, indicating that jointly learning the main and sub-category tasks through a shared representation benefits coarse-grained classification.

For Level 2 classification, the Hierarchy-Aware configuration produced the best overall results. AraBERTv2-base achieved the highest sub-category accuracy of 83.04% and a Macro F1-score of 0.7607, outperforming the corresponding Two-Stage, Bottom-Up, and Single Multi-Task variants.

Table 3. Performance comparison of different approaches on main class L_1 classification (Encoder). Bold values indicate the best performance in each column.

Approach	Model	Accuracy	Precision	Recall	Macro-F1
Single Multi-Task	AraBERTv2-base	0.9957	0.9951	0.9961	0.9956
	CAMeLBERT-MSA	0.9826	0.9835	0.9813	0.9823
	XLM-RoBERTa	0.9609	0.9589	0.9640	0.9605
	MARBERTv2	0.9783	0.9775	0.9785	0.9780
Two-Stage	AraBERTv2-base	0.9826	0.9815	0.9834	0.9842
	CAMeLBERT-MSA	0.9739	0.9735	0.9735	0.9735
	XLM-RoBERTa	0.9609	0.9592	0.9619	0.9604
	MARBERTv2	0.9696	0.9724	0.9766	0.9689
Bottom-Up (Flat)	AraBERTv2-base	0.9783	0.9797	0.9763	0.9779
	CAMeLBERT-MSA	0.9826	0.9823	0.9823	0.9823
	XLM-RoBERTa	0.9652	0.9634	0.9668	0.9648
	MARBERTv2	0.9652	0.9639	0.9658	0.9648
Hierarchy-Aware	AraBERTv2-base	0.9913	0.9912	0.9912	0.9912
	CAMeLBERT-MSA	0.9870	0.9873	0.9862	0.9867
	XLM-RoBERTa	0.9609	0.9592	0.9619	0.9604
	MARBERTv2	0.9870	0.9873	0.9862	0.9867

Table 4. Performance comparison of different approaches on sub-class L_2 classification (Encoder). Bold values indicate the best performance in each column.

Approach	Model	Accuracy	Precision	Recall	Macro-F1
Single Multi-Task	AraBERTv2-base	0.8043	0.6337	0.6520	0.6293
	CAMeLBERT-MSA	0.7913	0.6277	0.6521	0.6281
	XLM-RoBERTa	0.4957	0.1544	0.1831	0.1546
	MARBERTv2	0.7087	0.3570	0.3764	0.3532
Two-Stage	AraBERTv2-base	0.8130	0.6638	0.6353	0.6317
	CAMeLBERT-MSA	0.7700	0.5497	0.5646	0.5440
	XLM-RoBERTa	0.5913	0.2950	0.3028	0.2793
	MARBERTv2	0.6826	0.3600	0.3705	0.3492
Bottom-Up (Flat)	AraBERTv2-base	0.8174	0.6792	0.7022	0.6761
	CAMeLBERT-MSA	0.7652	0.6374	0.6326	0.6162
	XLM-RoBERTa	0.5783	0.2059	0.2446	0.2118
	MARBERTv2	0.7609	0.5911	0.5851	0.5676
Hierarchy-Aware	AraBERTv2-base	0.8304	0.7817	0.7701	0.7607
	CAMeLBERT-MSA	0.8174	0.6946	0.7098	0.6892
	XLM-RoBERTa	0.5957	0.2471	0.2780	0.2490
	MARBERTv2	0.7348	0.4191	0.4530	0.4221

Across all encoder-only experiments, Arabic-specialized models such as AraBERTv2-base and CAMeLBERT-MSA consistently outperformed the multilingual XLM-RoBERTa model, particularly in the fine-grained Level 2 classification task.

5.3. Encoder–Decoder Results (AraT5)

Tables 5 and 6 present the performance of the AraT5-based classification architectures on the held-out test set at both hierarchical levels.

The results demonstrate that the Hierarchy-Aware AraT5 configuration achieved the strongest overall performance within the encoder–decoder experiments across both hierarchical classification levels. At the fine-grained Level 2 task, the hierarchy-aware model achieved the highest accuracy (77.84%) and Macro-F1 score (50.43%), outperform-

ing all other AraT5-based architectures. Compared with the Bottom-Up (Flat) baseline, the Macro-F1 score improved from 48.04% to 50.43%, while the classification accuracy increased from 73.48% to 77.84%.

Table 5. AraT5: Level 1 (main class) performance comparison. Bold values indicate the best performance in each column.

Model	Accuracy	Precision	Recall	Macro-F1
Single Multi-Task	0.9870	0.9873	0.9862	0.9867
Two-Stage	0.9870	0.9873	0.9862	0.9867
Bottom-Up (Flat)	0.9783	0.9785	0.9774	0.9779
Hierarchy-Aware	0.9913	0.9912	0.9912	0.9912

Table 6. AraT5: Level 2 (sub-class) performance comparison. Bold values indicate the best performance in each column.

Model	Accuracy	Precision	Recall	Macro-F1
Single Multi-Task	0.7478	0.4282	0.4732	0.4349
Two-Stage	0.6696	0.2864	0.3312	0.2970
Bottom-Up (Flat)	0.7348	0.4651	0.5154	0.4804
Hierarchy-Aware	0.7784	0.5058	0.5298	0.5043

The Hierarchical Multi-Task architecture also achieved competitive performance, outperforming the Two-Stage pipeline across all Level 2 evaluation metrics. In contrast, the Two-Stage approach produced the weakest sub-class performance, which may be associated with cascading prediction errors propagated from the Level 1 stage to the downstream Level 2 classifier.

At the coarse-grained Level 1 task, all architectures achieved strong performance, with accuracy values exceeding 97%. Among the AraT5-based configurations, the Hierarchy-Aware model achieved the best Level 1 results, reaching 99.13% accuracy and a 99.12% Macro-F1 score.

Overall, the results suggest that incorporating hierarchical constraints during training may improve both coarse-grained and fine-grained legal case classification performance, particularly for challenging Level 2 sub-category prediction tasks. However, these findings should be interpreted in light of the experimental design, which relied on a single train/test split without repeated runs or cross-validation analyses. Future work will investigate the stability of the proposed approaches under additional evaluation protocols.

5.4. Prompt-Based Large Language Model Results

Table 7 presents the classification performance of GPT-4o under zero-shot, 3-shot, and 5-shot prompting configurations.

Under the zero-shot setting, GPT-4o achieved 96.52% L_1 accuracy and a Macro-F1 of 96.50%. At the fine-grained L_2 level, the model achieved 70.00% accuracy and a Macro-F1 of 69.64%, with an Exact Match Accuracy of 67.39% and a hierarchy compliance score of 96.52%.

Introducing in-context demonstrations improved performance across most evaluation metrics. Under the 3-shot configuration, L_1 accuracy increased to 98.26% and L_2 Macro-F1 improved to 72.68%. The 5-shot configuration achieved the strongest L_1 results (99.13% accuracy, 99.12% Macro-F1) and the highest L_2 Macro-F1 (74.30%), indicating improved performance on rare subclasses. However, L_2 accuracy was slightly higher under the 3-shot setting (71.74% vs. 71.30%), suggesting that additional demonstrations may shift the model toward better coverage of minority classes at a marginal cost to majority-class accuracy.

Hierarchy compliance improved progressively with additional demonstrations. The zero-shot configuration produced eight hierarchy violations, which decreased to two under 3-shot prompting and a single violation under the 5-shot setting, corresponding to a final compliance score of 99.57%.

Table 7. Classification performance of GPT-4o under different prompting strategies. Bold values indicate the best performance in each settings.

Level	Metric	Zero-Shot	Few-Shot (3)	Few-Shot (5)
L_1	Accuracy	0.9652	0.9826	0.9913
	Precision	0.9633	0.9835	0.9924
	Recall	0.9690	0.9813	0.9901
	Macro-F1	0.9650	0.9823	0.9912
L_2	Accuracy	0.7000	0.7174	0.7130
	Precision	0.7055	0.7403	0.7584
	Recall	0.7665	0.7918	0.8055
	Macro-F1	0.6964	0.7268	0.7430
Exact Match Accuracy		0.6739	0.7130	0.7130
Hierarchy Compliance		0.9652	0.9913	0.9957

The few-shot examples were intentionally selected based on error patterns observed during preliminary zero-shot experiments. This targeted strategy resulted in measurable improvements in both classification accuracy and structural consistency across the hierarchical label space.

Despite these improvements, supervised hierarchy-aware transformer models continued to achieve stronger fine-grained classification performance and fully consistent hierarchical predictions, particularly for challenging L_2 subclass prediction tasks.

5.5. Cross-Family Comparison

Table 8 summarizes the best-performing configuration from each methodological family at each hierarchical level. Since the architectures that achieved the highest Level 1 and Level 2 performance do not always coincide, the table reports each level's best result independently.

Table 8. Best-performing configuration from each methodological family at each hierarchical level. Bold values indicate the best performance in each column.

Family	Best Configuration	Acc.	Prec.	Recall	F1
Level 1 (Main Class)					
Encoder-Only	AraBERTv2/Single Multi-Task	0.9957	0.9951	0.9961	0.9956
Encoder-Decoder	AraT5v2/Hierarchy-Aware	0.9913	0.9912	0.9912	0.9912
LLM (GPT-4o)	5-Shot Prompting	0.9913	0.9924	0.9901	0.9912
Level 2 (Sub-Class)					
Encoder-Only	AraBERTv2/Hierarchy-Aware	0.8304	0.7817	0.7701	0.7607
Encoder-Decoder	AraT5v2/Hierarchy-Aware	0.7784	0.5058	0.5298	0.5043
LLM (GPT-4o)	5-Shot Prompting	0.7130	0.7584	0.8055	0.7430

At the coarse-grained L_1 task, the encoder-only AraBERTv2 under the Single Multi-Task architecture achieved the highest overall accuracy (99.57%) and Macro-F1 score (99.56%), outperforming both the best AraT5 and GPT-4o configurations. At the fine-grained L_2 task, AraBERTv2 under the Hierarchy-Aware architecture achieved the highest accuracy (83.04%) and Macro-F1 (76.07%), exceeding the best AraT5 configuration by

5.20 percentage points in accuracy and 25.64 points in Macro-F1. However, GPT-4o under 5-shot prompting achieved a competitive L_2 Macro-F1 of 74.30%—only 1.77 points below AraBERTv2—and the highest L_2 recall (80.55%) among all configurations, despite requiring no task-specific fine-tuning. GPT-4o also surpassed the fine-tuned AraT5 on L_2 Macro-F1 by over 23 percentage points, suggesting that large-scale pre-training and in-context learning can compensate for the absence of gradient-based optimization on this task.

Notably, the best L_1 and L_2 results within the encoder-only family were achieved by different architectures—Single Multi-Task and Hierarchy-Aware, respectively—suggesting that no single architecture simultaneously optimizes both hierarchical levels. This tradeoff is further examined in the Discussion (Section 6).

6. Discussion

This section examines the key findings from the experimental evaluation. We first analyze the effect of hierarchical modeling on classification performance, then compare multi-task and sequential training strategies. We subsequently discuss the differences between encoder-only and encoder–decoder architectures, the role of Arabic-specialized pre-training, and the impact of dataset distribution on model behavior. Finally, we present a detailed error analysis of both the best fine-tuned model and GPT-4o.

6.1. Effect of Hierarchical Modeling

The experimental results indicate that explicitly modeling hierarchical relationships improves fine-grained legal classification, particularly at the Level 2 subclass prediction task. The Hierarchy-Aware architecture achieved the highest Level 2 accuracy and Macro-F1 scores across both encoder-only and encoder–decoder families (Table 8), consistently outperforming the Bottom-Up (Flat) and Two-Stage alternatives.

Two complementary mechanisms contribute to this improvement. First, the hierarchy-consistency penalty (Equation (2)) discourages the model from assigning probability mass to Level 2 subclasses that are structurally incompatible with the correct Level 1 category during training, effectively regularizing the learned representations toward taxonomically coherent predictions. Second, the hard masking of invalid parent–child combinations at inference time eliminates logically inconsistent outputs entirely, guaranteeing full hierarchy compliance. Together, these mechanisms constrain the prediction space to semantically valid subclasses within each legal branch—a property that is especially valuable given the overlapping terminology shared by several criminal and civil subclasses in the Arabic legal taxonomy.

In contrast, the Bottom-Up (Flat) approach treats all 63 subclasses as independent labels, relying entirely on implicit co-occurrence patterns to recover Level 1 structure. While this baseline achieved competitive Level 2 accuracy in some configurations, it offers no structural guarantee that predictions will respect the legal taxonomy, making it less suitable for operational legal classification systems.

6.2. Multi-Task Learning vs. Sequential Classification

The results reveal a complementary relationship between the Single Multi-Task and Hierarchy-Aware architectures. The Single Multi-Task configuration achieved the strongest Level 1 performance (AraBERTv2: 99.57% accuracy), while the Hierarchy-Aware approach produced the best Level 2 results (AraBERTv2: 83.04% accuracy). This suggests a tradeoff between coarse-grained and fine-grained optimization across the evaluated architectures.

The Level 1 advantage of the Multi-Task configuration may be related to its joint optimization strategy, in which both classification levels are learned simultaneously through a shared encoder. Although the main-class objective receives only 20% of the total training

loss (Equation (1)), the shared optimization process appears sufficient for learning broad legal distinctions between criminal and civil cases. However, the same shared objective may reduce the model's ability to specialize in fine-grained subclass discrimination compared with architectures that explicitly enforce hierarchical consistency through dedicated structural constraints.

The Two-Stage pipeline consistently produced the weakest Level 2 performance across both transformer families. This behavior is likely associated with cascading errors, where an incorrect Level 1 prediction routes the input to the wrong second-stage classifier, making correct Level 2 prediction impossible regardless of the downstream classifier quality. In the encoder-only experiments, the Two-Stage AraBERTv2 achieved 81.30% Level 2 accuracy compared with 83.04% for the Hierarchy-Aware variant—a gap that becomes substantially larger in the AraT5 experiments (66.96% vs. 77.84%). The greater sensitivity of AraT5 to cascading errors may be related to the encoder–decoder architecture being less specialized for discriminative classification tasks compared with encoder-only models.

6.3. Encoder-Only vs. Encoder–Decoder Architectures

Encoder-only models consistently outperformed the AraT5 encoder–decoder configurations, with the performance gap most pronounced at Level 2. Under the same Hierarchy-Aware architecture, AraBERTv2 achieved a Macro-F1 of 76.07% compared with 50.43% for AraT5—a difference of over 25 percentage points.

This gap likely reflects a fundamental architectural distinction. Encoder-only models are pre-trained with a masked language modeling objective that optimizes contextualized token representations for downstream discriminative tasks. Classification decisions are derived directly from the [CLS] token, which provides a compact, task-oriented summary of the input. AraT5, by contrast, is pre-trained with a sequence-to-sequence objective oriented toward text generation. Although its encoder can be repurposed for classification, the representations it produces are optimized for decoder-side reconstruction rather than category separation.

Additionally, the Macro-F1 gap (25.64 points) is substantially larger than the accuracy gap (5.20 points), indicating that AraT5 struggles disproportionately with low-frequency subclasses. This is consistent with the long-tail distribution of the dataset (Figure 3): encoder-only models appear to generalize more effectively to rare categories, possibly because their discriminative pre-training objective encourages sharper decision boundaries even with limited examples.

6.4. Arabic-Specialized vs. Multilingual Models

Arabic-specialized models consistently outperformed XLM-RoBERTa across all four classification architectures. The gap was most pronounced at Level 2: under the Hierarchy-Aware architecture, AraBERTv2 achieved 83.04% accuracy versus 59.57% for XLM-RoBERTa—a difference of over 23 percentage points.

This performance difference likely stems from two factors. First, Arabic-specialized models are pre-trained on corpora that are predominantly or exclusively Arabic, allowing them to develop richer representations of Arabic morphology, including the root-pattern system, clitic attachment, and diacritization variation that characterize legal texts. Second, multilingual models must distribute their fixed model capacity across 100 languages, inherently limiting the depth of representation available for any single language. For a domain such as legal classification, where fine-grained distinctions often depend on subtle morphological or terminological cues, this capacity dilution appears to be a significant disadvantage.

The results also carry a practical implication: the additional computational cost of XLM-RoBERTa's larger architecture did not translate into improved classification performance. Arabic-specialized models achieved stronger results while requiring shorter training times, suggesting that targeted linguistic adaptation is more cost-effective than broader multilingual coverage for domain-specific Arabic NLP tasks.

6.5. Impact of Dataset Distribution

The long-tail distribution of Level 2 subclasses (Figure 3) is a key factor underlying the performance gap between Level 1 and Level 2 classification across all experiments. While a small number of dominant subclasses are well represented in the training data, many categories contain relatively few examples, making robust decision boundary learning difficult.

This imbalance interacts differently with each classification architecture. The Two-Stage pipeline is particularly affected: partitioning the training data into separate second-stage classifiers further reduces the number of examples available for rare subclasses within each branch, compounding the data scarcity problem. In contrast, the single-model architectures (Multi-Task, Hierarchy-Aware, and Bottom-Up) train on the full dataset simultaneously, preserving exposure to all subclasses and enabling better generalization to low-frequency categories.

The imbalance also partially explains the large gap between accuracy and Macro-F1 observed in several configurations. Because accuracy is dominated by high-frequency classes, models that perform well on common subclasses can achieve reasonable accuracy while failing on rare ones. The Macro-F1 metric, which weights all classes equally, reveals these failures more clearly—as seen in the AraT5 Two-Stage configuration, where accuracy reached 66.96% but Macro-F1 dropped to 29.70%.

6.6. Error Analysis

A class-level analysis of the best-performing Hierarchy-Aware AraBERTv2 model revealed substantial variation across Level 2 subclasses. Table 9 presents the five highest- and five lowest-performing subclasses by F1-score, restricted to classes with at least five test samples.

Table 9. Top-5 and bottom-5 Level 2 subclasses by F1-score (AraBERTv2/Hierarchy-Aware, support ≥ 5).

Sub-Class	Precision	Recall	F1	Support
Top 5				
نفقة (Alimony)	1.000	1.000	1.000	9
سرقة (Theft)	1.000	0.923	0.960	13
إيذاء الغير (Harming Others)	0.889	1.000	0.941	8
رد مسروق (Return of Stolen Goods)	0.857	1.000	0.923	6
فسخ نكاح (Marriage Annulment)	0.857	1.000	0.923	6
Bottom 5				
اعتداء (Assault)	0.667	0.500	0.571	8
تعاطي (Drug Use)	0.800	0.667	0.727	6
عقار (Real Estate)	0.667	0.800	0.727	5
مطالبة بقيمة أتعاب (Fee Claims)	1.000	0.600	0.750	5
بيع (Sale)	0.722	0.867	0.788	15

The strongest performance was observed for subclasses such as نفقة (Alimony), سرقة (Theft), and فسخ نكاح (Marriage Annulment), all of which achieved F1-scores above

0.92. These categories are characterized by relatively distinctive legal terminology and well-defined legal concepts, making them easier to separate from competing subclasses.

In contrast, اعتداء (Assault) achieved the lowest F1-score (0.571), with only half of its test instances correctly recalled. Other low-performing subclasses, including تعاطي (Drug Use, F1: 0.727), عقار (Real Estate, F1: 0.727), and بيع (Sale, F1: 0.788), share substantial terminology, legal context, or procedural characteristics with related classes, increasing the difficulty of fine-grained discrimination. Notably, مطالبة بقيمة أتعاب (Fee Claims) achieved perfect precision but only 60% recall, indicating that the model is conservative in predicting this category—when it does, it is always correct, but it frequently assigns these cases to broader civil categories instead.

The confusion analysis further supports this observation. The most frequent errors occurred between legally related subclasses rather than between unrelated legal domains. المخدرات والمسكرات (Drug Possession) was misclassified as ترويج مخدرات (Drug Trafficking) in 3 out of 26 test instances and as تهريب مخدرات (Drug Smuggling) in 2 instances, reflecting the shared vocabulary across categories that describe different stages of the same criminal activity chain. Similarly, اعتداء (Assault) was confused with قتل (Homicide) and سلاح (Weapons), while civil categories such as بيع (Sale), إجارة (Lease), and عقار (Real Estate) were occasionally confused with one another.

Notably, the most frequently confused subclasses are not necessarily the rarest. والمسكرات (Drug Possession) and ترويج مخدرات (Drug Trafficking) are the two most represented classes in the dataset (Figure 3), yet Drug Possession produced the highest number of misclassifications. Similarly, اعتداء (Assault) achieved the lowest F1-score despite having adequate training representation. This indicates that terminological and semantic overlap between legally related subclasses poses a greater classification challenge than data scarcity alone, and that increasing the number of training examples may not resolve confusions rooted in genuine domain ambiguity.

These findings suggest that the primary classification challenge lies not in distinguishing between broad legal domains, but in separating closely related subclasses that share substantial legal terminology and factual characteristics. The rarity of cross-domain confusions indicates that the hierarchy-aware model captures high-level legal distinctions successfully, while the remaining errors are concentrated around fine-grained boundary cases within the same legal branch. This observation is consistent with both the hierarchical nature of the legal taxonomy and the long-tail distribution of the dataset.

7. Conclusions

This paper presented a systematic investigation of hierarchical multi-class classification of Arabic legal text using a novel dataset of 1146 judicial cases sourced from the Saudi Ministry of Justice. We evaluated three methodological families—encoder-only transformers, encoder-decoder transformers, and prompt-based large language models—under four classification architectures, providing the first comprehensive benchmark of this kind on Arabic judicial records.

The central finding is that explicit hierarchical supervision, realized through the Hierarchy-Aware architecture with a consistency penalty and inference-time logit masking, yields the strongest and most structurally coherent predictions across all evaluated model families. For encoder-only models, AraBERTv2 with Hierarchy-Aware training achieved the best overall L_2 performance (83.04% accuracy, 76.07% Macro-F1), while the AraT5 Hierarchy-Aware variant led among encoder-decoder models (77.84% accuracy, 50.43% Macro-F1). GPT-4o under 5-shot prompting achieved 99.13% L_1 accuracy and a competitive L_2 Macro-F1 of 74.30%—only 1.77 points below the best fine-tuned model—without any task-specific training. However, supervised hierarchy-aware models maintained stronger

overall fine-grained performance and guaranteed hierarchical consistency, indicating that task-specific fine-tuning on domain-adapted representations retains a meaningful advantage for structured legal classification tasks.

Several directions remain for future work. First, the full judgment text—excluded here due to length constraints—could be incorporated through long-document architectures such as Longformer or hierarchical chunk encoders, potentially providing richer contextual signals for rare sub-classes. Second, data augmentation strategies tailored to low-resource legal sub-classes, including back-translation, paraphrasing, or retrieval-augmented generation, could mitigate the 43:1 imbalance ratio that limits fine-grained classification performance. Third, prompt-based legal classification could be enhanced through retrieval-augmented generation (RAG), dynamically incorporating jurisdiction-specific statutes, judicial precedents, or other external legal knowledge during inference. Such an approach may improve prompt-based reasoning while preserving the generality of the classification framework. Fourth, the dataset could be extended to include additional Saudi judicial domains and expanded to cover other Arabic-speaking legal systems, enabling cross-jurisdictional evaluation. Finally, the hierarchical taxonomy could be deepened to a third classification level—for instance, distinguishing specific Shari’a provisions within each sub-class—to enable more granular legal document retrieval and analysis.

Author Contributions: Conceptualization, N.A. (Nourah Alangari), N.A. (Nouf Alshenaifi) and H.A.; methodology, N.A. (Nourah Alangari), N.A. (Nouf Alshenaifi) and H.A.; software, N.A. (Nourah Alangari), N.A. (Nouf Alshenaifi) and H.A.; validation, N.A. (Nourah Alangari), N.A. (Nouf Alshenaifi) and H.A.; formal analysis, N.A. (Nourah Alangari); data curation, N.A. (Nourah Alangari), N.A. (Nouf Alshenaifi) and H.A.; writing—original draft preparation, N.A. (Nouf Alshenaifi); writing—review and editing, N.A. (Nourah Alangari), N.A. (Nouf Alshenaifi) and H.A.; visualization, N.A. (Nouf Alshenaifi). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. GPT-4o Classification Prompt

Prompt Used for Hierarchical Legal Case Classification

The prompt in Figure A1 was used for the GPT-4o experiments. The model was instructed to perform hierarchical legal case classification by predicting both the main class (Criminal or Civil) and a valid sub-class constrained by the predefined taxonomy. The model was required to return the output strictly in JSON format.

For readability and to facilitate reproducibility, an English translation of the Arabic system prompt used in the experiments is provided below. This translation is included for explanatory purposes only; all experiments were conducted using the original Arabic prompt.

System Prompt (English Translation)

You are a legal case classifier.

Your task is to classify each legal case into two hierarchical levels:

1. Main_Class:
 - Criminal
 - Civil

2. Sub_Class: You must select the sub-class only from the list of allowed categories corresponding to the selected main class.

Important rules:

- Do not select a Sub_Class that is not included in the allowed list.
- The selected Sub_Class must belong to the corresponding Main_Class.
- If the case involves a crime, punishment, conviction, or discretionary criminal penalty (Ta'zir), it is generally classified as Criminal.
- If the case concerns a private claim such as money, contracts, alimony, child custody, compensation, sale, or real estate, it is generally classified as Civil.
- If a crime is mentioned only as background context for a civil claim, classify the case as Civil.

Output format:

```
{
  "Main_Class": "Criminal or Civil",
  "Sub_Class": "one valid category from the allowed list"
}
```

SYSTEM PROMPT

أنت مصنف قضايا قانونية.

المطلوب تصنيف كل قضية إلى مستويين:

Main_Class (1)

- جنائي
- حقوقي

Sub_Class (2)

يجب أن تختار التصنيف الفرعي فقط من القائمة المسموحة حسب التصنيف الرئيسي.

القوائم المسموحة:

جنائي: VALID_CLASSES ["جنائي"]

حقوقي: VALID_CLASSES ["حقوقي"]

قواعد مهمة:

- لا تختار Sub_Class غير موجود في القائمة.
- يجب أن يكون Sub_Class تابعاً لـ Main_Class الصحيح.
- إذا كانت القضية تتضمن جريمة وعقوبة أو إدانة أو تعزيز فهي غالباً جنائية.
- إذا كانت مطالبة خاصة مثل مال، عقد، نفقة، حضانة، تعويض، بيع، عمار فهي غالباً حقوقي.
- إذا ذكرت الجريمة فقط كسياق لمطالبة مدنية، صنفها حقوقي.

أجب بصيغة JSON فقط بدون أي شرح:

```
{
  "Main_Class": "جنائي أو حقوقي",
  "Sub_Class": "واحد فقط من القائمة"
}
```

Figure A1. Arabic system prompt used in the GPT-4o experiments for hierarchical legal case classification. The prompt constrains the model to predict a valid main class and a corresponding sub-class from the predefined taxonomy and to return the output in JSON format.

Appendix B. Complete Parent–Child Mapping of the Legal Case Taxonomy

This appendix presents the hierarchical taxonomy used in the experiments. The classification framework consists of two Level-1 (L_1) categories (Criminal and Civil) and a set of Level-2 (L_2) subclasses. For readability, Table A1 reports only subclasses with at least 10 instances, which constitute the majority of the dataset. The parent–child relationships reported here were used in the hierarchy-aware classification model, masking mechanism, and GPT-4o prompting strategy.

Table A1. Level-2 (L_2) subclass taxonomy used in the experiments. Only subclasses with at least 10 instances are shown for readability. The table explicitly shows the parent–child mapping between the two Level-1 (L_1) categories and their corresponding Level-2 subclasses.

L1 Parent	Arabic Subclass	English Translation	Count
Criminal	المخدرات والمسكرات (حيازة)	Narcotics Possession	131
	ترويج مخدرات	Drug Trafficking	92
	سرقة	Theft	63
	شرب مسكر	Drinking Intoxicants	50
	اعتداء	Assault	38
	إيذاء الغير	Harming Others	38
	تعاطي	Drug Consumption	32
	دية	Blood Money (Diyya)	20
	تهريب مخدرات	Drug Smuggling	15
	قتل	Homicide	15
	زنا	Zina (Fornication)	12
	تحرش	Harassment	11
	حرابة	Hiraba (Banditry)	11
	جرائم معلوماتية	Cybercrimes	10
	أرش إصابة	Injury Compensation (Arsh)	10
Civil	بيع	Sale	73
	نفقة	Alimony/Maintenance	45
	إجارة	Lease/Tenancy	44
	اختصاص	Jurisdiction	38
	فسخ نكاح	Annulment of Marriage	29
	رد مسروق	Return of Stolen Property	28
	مطالبة بقيمة أتعاب	Claim for Professional Fees	26
	عقار	Real Estate	25
	شراكة ومضاربة	Partnership and Speculation	21
	حضانة	Child Custody	20
	قرض	Loan	18
	إعسار	Insolvency	17
	كفالة	Suretyship	15
	حادث مروري	Traffic Accident	13
	قسمة تركة	Division of Inheritance	11
	رد محتلس	Return of Embezzled Property	10

References

- Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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*; Burstein, J., Doran, C., Solorio, T., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; Volume 1, pp. 4171–4186. [CrossRef]
- Chalkidis, I.; Fergadiotis, M.; Androutsopoulos, I. Multieurlex-a multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 6974–6996.
- Chen, H.; Wu, L.; Chen, J.; Lu, W.; Ding, J. A comparative study of automated legal text classification using random forests and deep learning. *Inf. Process. Manag.* **2022**, *59*, 102798. [CrossRef]

4. González-González, J.; de Arriba-Pérez, F.; García-Méndez, S.; Busto-Castiñeira, A.; González-Castaño, F.J. Automatic explanation of the classification of Spanish legal judgments in jurisdiction-dependent law categories with tree estimators. *J. King Saud. Univ.-Comput. Inf. Sci.* **2023**, *35*, 101634. [[CrossRef](#)]
5. Tomar, M.S.; Gupta, V. Legal case classification using machine learning with NLP. In *Proceedings of the 2023 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAIAI)*; IEEE: New York, NY, USA, 2023; pp. 1–6.
6. Prasad, N.; Boughanem, M.; Dkaki, T. Effect of hierarchical domain-specific language models and attention in the classification of decisions for legal cases. In *Proceedings of the CIRCLE (Joint Conference of the Information Retrieval Communities in Europe)*, Samatan, France, 4–7 July 2022.
7. Qin, R.; Huang, M.; Liu, J.; Miao, Q. Hybrid attention-based transformer for long-range document classification. In *Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN)*; IEEE: New York, NY, USA, 2022; pp. 1–8.
8. Song, D.; Vold, A.; Madan, K.; Schilder, F. Multi-label legal document classification: A deep learning-based approach with label-attention and domain-specific pre-training. *Inf. Syst.* **2022**, *106*, 101718. [[CrossRef](#)]
9. Tuggenier, D.; Von Däniken, P.; Peetz, T.; Cieliebak, M. LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*; European Language Resources Association: Paris, France, 2020; pp. 1235–1241.
10. Xu, S.; Stauffer, L.; Santosh, T.Y.S.S.; Ichim, O.; Heri, C.; Grabmair, M. Vechr: A dataset for explainable and robust classification of vulnerability type in the european court of human rights. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 11738–11752.
11. Papaloukas, C.; Chalkidis, I.; Athinaios, K.; Pantazi, D.; Koubarakis, M. Multi-granular legal topic classification on Greek legislation. In *Proceedings of the Natural Legal Language Processing Workshop 2021*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 63–75.
12. Sargeant, H.; Izzidien, A.; Steffek, F. Topic classification of case law using a large language model and a new taxonomy for UK law: AI insights into summary judgment. *Artif. Intell. Law* **2025**, *34*, 443–491. [[CrossRef](#)]
13. Vatsal, S.; Meyers, A.; Ortega, J.E. Classification of us supreme court cases using bert-based techniques. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*; INCOMA Ltd.: Shoumen, Bulgaria, 2023; pp. 1207–1215.
14. Kementchedjhieva, Y.; Chalkidis, I. An exploration of encoder-decoder approaches to multi-label classification for legal and biomedical text. In *Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 5828–5843.
15. Dominguez-Olmedo, R.; Nanda, V.; Abebe, R.; Bechtold, S.; Engel, C.; Frankenreiter, J.; Gummadi, K.; Hardt, M.; Livermore, M. Lawma: The Power of Specialization for Legal Annotation. In *Proceedings of the Thirteenth International Conference on Learning Representations. International Conference on Learning Representations, Singapore, 24–28 April 2025*; pp. 1–45.
16. Weichbroth, P. Ai and the law: Evaluating chatgpt’s performance in legal classification. In *Proceedings of the European Conference on Artificial Intelligence*; Springer: Cham, Switzerland, 2025; pp. 69–81.
17. Ikram, A.Y.; Chakir, L. Arabic text classification in the legal domain. In *Proceedings of the 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*; IEEE: New York, NY, USA, 2019; pp. 1–6.
18. Al-qurishi, M.; Alqaseemi, S.; Souissi, R. AraLegal-BERT: A pretrained language model for Arabic Legal text. In *Proceedings of the Natural Legal Language Processing Workshop 2022*; Aletras, N., Chalkidis, I., Barrett, L., Goanță, C., Preotiuc-Pietro, D., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 338–344. [[CrossRef](#)]
19. Bouhouche, A.; Esghir, M.; Errachid, M. Classification of Moroccan Legal and Legislative Texts Using Machine Learning Models. *Int. J. Adv. Comput. Sci. Appl.* **2024**, *15*, 1108. [[CrossRef](#)]
20. Ammar, A.; Koubaa, A.; Benjdira, B.; Nacar, O.; Sibae, S. Prediction of arabic legal rulings using large language models. *Electronics* **2024**, *13*, 764. [[CrossRef](#)]
21. Tahtah, O.; Zinedine, A.; Fardousse, K. Arabic Legal Question Classification Using Deep Learning and Adapted Word Embedding. *Proc. Oper. Res. Forum* **2025**, *6*, 112. [[CrossRef](#)]
22. Antoun, W.; Baly, F.; Hajj, H. AraBERT: Transformer-based Model for Arabic Language Understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*; European Language Resources Association: Paris, France, 2020.
23. Inoue, G.; Alhafni, B.; Baimukan, N.; Bouamor, H.; Habash, N. The Interplay of Variant, Size, and Task Type in Arabic Pre-trained Language Models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021.
24. Abdul-Mageed, M.; Elmadany, A.; Nagoudi, E.M.B. ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 7088–7105.

25. Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 8440–8451.
26. Nagoudi, E.M.B.; Elmadany, A.; Abdul-Mageed, M. AraT5: Text-to-text transformers for Arabic language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; Volume 1, pp. 628–647.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.