Legal Textual Entailment

1st Hiba Abrar
MS AI 2k23
National University of
Science & Technology
(NUST), Pakistan

2nd Shaiza (00000533150)
PHD-EE 2k24
National University of
Science & Technology
(NUST), Pakistan

3rd Sijjil Shabbir
MSEE 2k23
National University of
Science & Technology
(NUST), Pakistan

***Abstract*—This project addresses the legal entailment task using data from the COLIEE competition, focusing on Japanese statutory law. Due to the limited availability of high-performing legal language models for Japanese and the complexity of Japanese legal text, the dataset was translated into English for processing. The training data, originally in XML format, was parsed and consolidated into a CSV file containing approximately 1,200 article–hypothesis–label examples. We initially approached the task using an embedding-based semantic similarity framework by fine-tuning a SentenceTransformer initialized with the nlpaueb/legal-bert-base-uncased model. Despite implementing Top-K layer fine-tuning and gradual unfreezing to mitigate overfitting, the model failed to produce satisfactory results due to the dataset's limited size and the model's large parameter count. Subsequently, we adopted a more lightweight model, legal-bert-small-uncased, and repeated the embedding-based approach. However, performance improvements remained marginal. As an alternative, we restructured the task as a binary classification problem, treating each article-query pair as a natural language inference input using standard BERT tokenization and cross-entropy loss. Fine-tuning legal-bert-small-uncased under this formulation yielded better alignment with entailment labels, offering a more robust solution for structured legal reasoning in a low-resource setting. Our findings emphasize the importance of task formulation and model size when applying transformer-based models to domain-specific entailment problems.**

***Index Terms*—Legal Entailment, COLIEE, LegalBERT, Sentence Transformers, Top-K Fine-Tuning, Natural Language Inference, Legal NLP, Japanese Law, Embedding Similarity, Cross-Entropy Loss, Transfer Learning, Domain Adaptation**

I. Introduction

Legal Textual Entailment (LTE) poses a considerable challenge in the realms of natural language processing (NLP) and artificial intelligence, especially when applied to the legal field. Assessing whether a specific legal proposition (hypothesis or query) can be logically deduced from another legal document (premise or article) requires a deep understanding of intricate legal terminology, statutes, and reasoning. Automating this task with precision has significant potential to improve legal research, case analysis, and overall productivity of legal professionals. The COLIEE (Competition on Legal Information Extraction and Entailment) serves as an essential standard for assessing systems developed to address these challenges. In particular, COLIEE 2025 emphasizes tasks aimed at retrieving relevant legal articles (Task 3) and then analyzing entailment relationships between a query and the retrieved articles (Task 4). An essential criterion of the competition is the creation of fully automated systems that can carry out these tasks independently, without any human involvement. This paper outlines a system created to tackle the COLIEE 2025 LTE challenge, incorporating both retrieval and entailment aspects as executed in the accompanying computational notebook. The system utilizes a two-phase pipeline architecture. The initial stage involves retrieving information by identifying relevant articles from the Japanese Civil Code in response to a specific legal query. This is accomplished by creating dense vector embeddings for all civil code articles using a pre-trained language model tailored for the legal field, specifically nlpaueb/legal-bert-base-uncased [1]. Queries are also embedded, and cosine similarity is applied to rank and select the top candidate articles most relevant to the query. The core entailment task is tackled in the second stage. The retrieved articles and the initial query are input into a sequence classification model, utilizing the fine-tuned nlpaueb/legal-bert-base-uncased model once again. This model is designed to determine if an entailment relationship (Yes/No) exists between the article and the query. To enhance performance and adapt the general-purpose language model to the specific nuances of legal entailment, several techniques were employed. This involves integrating engineered lexical features (like token overlap and length ratios) with transformer-based embeddings, applying a layer-freezing technique during fine-tuning (by freezing the first 8 layers of BERT) to maintain the core understanding of legal language while modifying the upper layers, and adjusting the classification threshold according to validation set results (particularly aiming to maximize the F1 score) instead of using a standard 0.5 threshold. The system employs the RITE (Recognizing Inference in TExt) dataset format supplied by COLIEE for both training and assessment. This introduction describes the problem setting, the competition structure, and the overall architecture and main approaches of the deployed LTE system.

II. Literature Review

Textual entailment, or natural language inference (NLI), is a key task in natural language processing (NLP) that involves determining whether a particular hypothesis logically follows from a given premise. In the legal domain, this task is critical for automating legal reasoning, verifying case consistency, retrieving relevant precedents, and assisting in contract analysis. The complexity of legal language, characterized by long sentences, domain-specific terminology, and formal structure, makes legal textual entailment (LTE) a particularly challenging problem.

Early approaches to textual entailment were primarily rule-based or employed traditional machine learning algorithms. These models relied on features such as lexical overlap, dependency parse trees, and syntactic patterns. Support Vector Machines (SVM), Decision Trees, and Naïve Bayes classifiers using bag-of-words or TF-IDF representations were common during this era. However, these shallow methods often failed to capture the deeper semantic relationships needed to infer entailment, especially in legal texts where linguistic nuance plays a significant role [4].

The advent of deep learning shifted the landscape dramatically. Recurrent Neural Networks (RNNs), and particularly Long Short-Term Memory (LSTM) networks, were able to model sequences of words and learn contextual relationships. Despite their advances over traditional approaches, these models were still unable to handle the lengthy dependencies found in legal texts. An important turning point was when Vaswani et al. introduced the Transformer architecture in 2017, which allowed models to analyze text in parallel and use attention methods to capture both short- and long-range dependencies [5].

Pre-trained transformer models such as BERT (Bidirectional Encoder Representations from Transformers), RoBERTa, and ALBERT demonstrated remarkable performance on general NLI tasks, particularly on datasets like SNLI and MNLI. The capacity of BERT to produce contextual embeddings for every token transformed entailment and other downstream NLP tasks. Yet, because of the disparities in vocabulary and linguistic structure, general-purpose models trained on Wikipedia and BooksCorpus data did not function as well when applied directly to legal texts [6].

Recognizing this domain mismatch, researchers introduced transformer-based models fine-tuned or pre-trained on legal corpora. Legal-BERT, introduced by Chalkidis et al., was one of the first models tailored for the legal domain. Trained on a wide range of legal documents including European legislation, court cases, and contracts, Legal-BERT consistently outperformed standard BERT in legal text classification, semantic similarity, and entailment tasks [5]. Another important contribution was CaseLaw-BERT, trained specifically on U.S. case law data, which enhanced performance in tasks related to judicial decisions and precedent analysis [7].

A number of datasets have been created to aid legal entailment research concurrently with model developments. A standard in this field is the COLIEE (Competition on Legal Information Extraction/Entailment) dataset, which provides annotated legal text pairs from Japanese case law and bar exams. It poses entailment problems at the phrase level that have practical applications [6]. Fine-grained entailment and argument mining are made possible by the European Court of Human Rights (ECtHR) collection, which offers thousands of cases and their legal justifications [6]. Furthermore, labeled legal clauses from commercial contracts are included in the CUAD (Contract Understanding Atticus Dataset) and can be used for entailment research [9].

Large language models (LLMs) like GPT-3, GPT-4, and LLaMA, which have shown few-shot and zero-shot capabilities on entailment tasks, are used in more recent work. Even while these models demonstrate potential in general thinking, legal circumstances still require domain-specific fine-tuning. To further improve the accuracy and transparency of entailment predictions in delicate legal situations, some new works investigate merging LLMs with retrieval-based techniques to ground predictions in statutory text or legal precedent [6].

In conclusion, rule-based approaches, deep neural networks, and now potent transformer-based models have all been used in the field of legal text entailment. The most important element in attaining excellent performance in legal NLP tasks has been shown to be domain adaptability. The precision and usefulness of entailment systems in actual legal practice are steadily increasing due to the expanding availability of legal-specific datasets and pre-trained models.

III. Methodology

The legal dataset used in this project was obtained from the COLIEE (Competition on Legal Information Extraction and Entailment) competition. The data pertains to Japanese laws. Since comprehending legal documents in Japanese proved challenging—and given that more pre-trained legal models are available in English than in Japanese—we opted to work with an English-translated version of the dataset. Although multilingual models are available on Hugging Face, they often lack proficiency in handling complex legal terminology, which further supported our decision to use English data.

The first step in the pipeline was translating the Japanese legal documents into English. The training dataset comprises 18 XML files. Each XML file contains multiple article–hypothesis pairs, along with a unique pair ID and a label indicating whether the article entails the hypothesis.

Next, we preprocessed these XML files, converting them into a single CSV file with four columns: id, article, query, and label. This yielded approximately 1,200 training examples. We then performed an 80:20 train-validation split.

For model selection, we chose nlpaueb/legal-bert-base-uncased, a LegalBERT variant with 12 transformer layers and 110 million parameters.

The project uses a sentence embedding-based similarity framework for the legal entailment. Specifically, fine-tuning is done for a pre-trained SentenceTransformer model using CosineSimilarityLoss, which optimizes the model to bring semantically similar sentence pairs (query and article) closer in the embedding space. For evaluation, the EmbeddingSimilarityEvaluator was utilized, where the entailment labels (”Y”/”N”) are mapped to binary scores (1/0), and cosine similarity between query-article pairs is used to assess alignment. This method allows the model to learn a dense vector space where entailment corresponds to higher similarity, facilitating robust semantic matching without direct classification. The overall workflow is illustrated in Fig. 1.

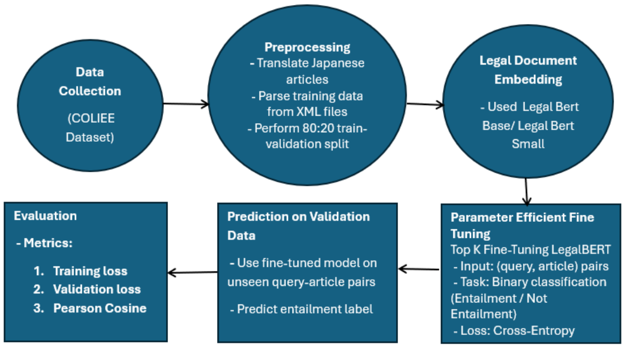

Data Collection
(COLIEE Dataset)
Preprocessing
- Translate Japanese articles
- Parse training data from XML files
- Perform 80:20 train-validation split
Legal Document Embedding
- Used Legal Bert Base/ Legal Bert Small
Parameter Efficient Fine Tuning
Top K Fine-Tuning LegalBERT
- Input: (query, article) pairs
- Task: Binary classification (Entailment / Not Entailment)
- Loss: Cross-Entropy
Prediction on Validation Data
- Use fine-tuned model on unseen query-article pairs
- Predict entailment label
Evaluation
- Metrics:
1. Training loss
2. Validation loss
3. Pearson Cosine

Fig. 1: Legal Entailment Task WorkFlow

Initially, we fine-tuned the entire model on the training dataset. However, due to the model's large size and the relatively small training set, this led to significant overfitting.

To address this, we adopted a Top-K layer fine-tuning strategy along with other updates like lexical overlap. We also tried to implement an end-to-end pipeline on the given Japanese dataset without translating it into English. In the first attempt, we fine-tuned only the top 2 layers of the model while freezing the remaining 10 layers. We also tried to freeze 8 layers and train the top 4 layers. However, this did not yield satisfactory results. We then implemented a gradual unfreezing strategy, successively unfreezing the top layers one by one: first the top 3 layers, then the top 4, and so on, up to the top 6 layers. Each configuration was fine-tuned for 10 epochs.

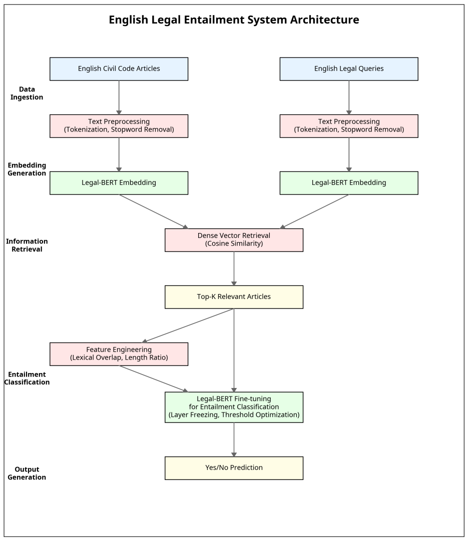

English Legal Entailment System Architecture
Data Ingestion
English Civil Code Articles
English Legal Queries
Text Preprocessing (Tokenization, Stopword Removal)
Text Preprocessing (Tokenization, Stopword Removal)
Embedding Generation
Legal-BERT Embedding
Legal-BERT Embedding
Information Retrieval
Dense Vector Retrieval (Cosine Similarity)
Top-K Relevant Articles
Feature Engineering (Lexical Overlap, Length Ratio)
Entailment Classification
Legal-BERT Fine-tuning for Entailment Classification (Layer Freezing, Threshold Optimization)
Output Generation
Yes/No Prediction

Fig. 2: Legal Entailment Task Workflow (English) with freezing 8 layers and training top 4 layers only.

Despite these efforts, overfitting persisted at each stage, even with the Top-K approach. The result for the Top-K approach fine-tuning is shown in Fig. 4. The results of training loss along with the Pearson Cosine for each step is also listed in the Table I.

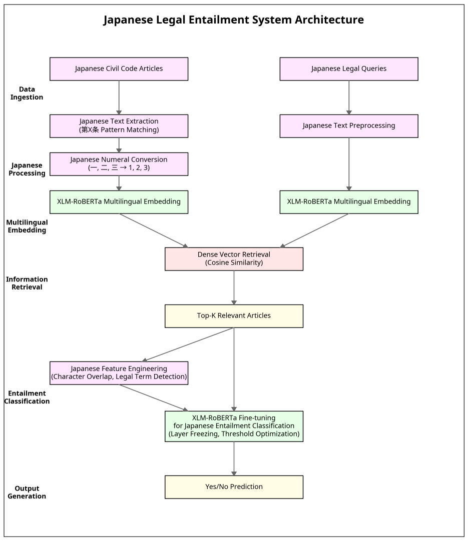

Japanese Legal Entailment System Architecture
Data Ingestion
Japanese Civil Code Articles
Japanese Legal Queries
Japanese Text Extraction (第X条 Pattern Matching)
Japanese Text Preprocessing
Japanese Processing
Japanese Numeral Conversion (一, 二, 三 → 1, 2, 3)
XLM-RoBERTa Multilingual Embedding
XLM-RoBERTa Multilingual Embedding
Multilingual Embedding
Dense Vector Retrieval (Cosine Similarity)
Information Retrieval
Top-K Relevant Articles
Japanese Feature Engineering (Character Overlap, Legal Term Detection)
Entailment Classification
XLM-RoBERTa Fine-tuning for Japanese Entailment Classification (Layer Freezing, Threshold Optimization)
Output Generation
Yes/No Prediction

Fig. 3: Legal Entailment Task Workflow (Japanese).

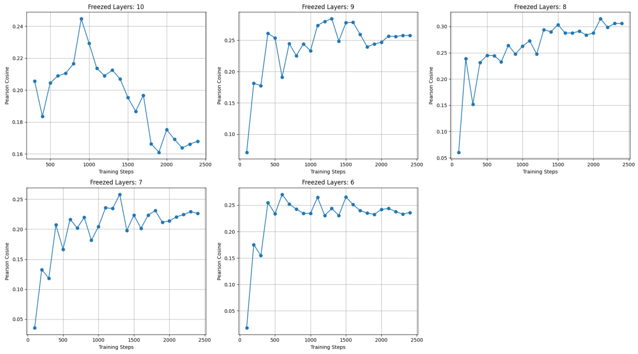


Fig. 4: Fine Tuning the Legal Bert Base with Top-K fine tuning

Since satisfactory results were not achieved using the LegalBERT base model, we switched to the smaller variant, nlpaueb/legal-bert-small-uncased. This model consists of 6 layers and approximately 35 million parameters. To mitigate overfitting, we again employed the Top-K fine-tuning approach. This time, the model was fine-tuned using the Top 1, Top 2, and Top 3 layers for 15 epochs. The results of this fine-tuning strategy with the LegalBERT small model are presented in Fig. 5. The results of fine-tuning using the SentenceTransformer-based approach, including training

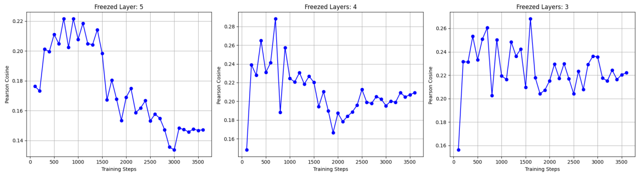


Fig. 5: Fine Tuning the Legal Bert Small with Top-K fine tuning

	Top 2 Layers		Top 3 Layers		Top 4 Layers		Top 5 Layers		Top 6 Layers	
Step	Training Loss	Pearson Cosine	Training Loss	Pearson Cosine	Training Loss	Pearson Cosine	Training Loss	Pearson Cosine	Training Loss	Pearson Cosine
100	0.3145	0.13245	0.3037	0.07126	0.2987	0.06061	0.2971	0.03574	0.2955	0.01794
200	0.2648	0.148211	0.2786	0.18136	0.2778	0.23862	0.2801	0.1327	0.2807	0.1753
300	0.2359	0.20553	0.238	0.1774	0.2424	0.15183	0.2438	0.11835	0.2474	0.15459
400	0.2276	0.1835	0.2199	0.26096	0.2259	0.23143	0.2222	0.20714	0.2358	0.2549
500	0.2105	0.20446	0.2118	0.25346	0.2226	0.24521	0.2204	0.16661	0.2281	0.23359
600	0.207	0.20888	0.1509	0.191	0.1535	0.24457	0.1419	0.21649	0.1494	0.27045
700	0.2028	0.21054	0.1611	0.24455	0.1658	0.23281	0.156	0.2021	0.1691	0.25225
800	0.1968	0.21647	0.1248	0.22524	0.1198	0.26421	0.1057	0.21954	0.1172	0.24249
900	0.1948	0.24472	0.1154	0.24404	0.1079	0.24764	0.0981	0.18154	0.1015	0.23454
1000	0.1881	0.22941	0.0986	0.23318	0.09	0.26254	0.0752	0.20434	0.0783	0.23445
1100	0.1603	0.21363	0.0761	0.27373	0.065	0.27269	0.0524	0.2356	0.0573	0.2647
1200	0.1611	0.20897	0.0772	0.27979	0.0696	0.24742	0.053	0.2344	0.0507	0.23052
1300	0.1336	0.21244	0.0569	0.28424	0.0436	0.29401	0.0312	0.2581	0.0314	0.24367
1400	0.1431	0.20701	0.0546	0.24862	0.0459	0.28993	0.0346	0.19792	0.0364	0.23042
1500	0.1257	0.19535	0.0433	0.2779	0.0284	0.30347	0.0221	0.22329	0.0214	0.26535
1600	0.1318	0.18667	0.0433	0.27853	0.0266	0.2878	0.0174	0.20141	0.0154	0.25101
1700	0.1308	0.19674	0.0392	0.25921	0.0262	0.28752	0.0216	0.22347	0.0224	0.23988
1800	0.1183	0.1663	0.0292	0.23946	0.017	0.29091	0.0118	0.23107	0.0089	0.23493
1900	0.1083	0.16102	0.0303	0.24394	0.0168	0.28368	0.0135	0.21177	0.0102	0.23264
2000	0.1048	0.17514	0.0272	0.24681	0.0146	0.28763	0.0107	0.2137	0.0081	0.2419
2100	0.0984	0.16911	0.0197	0.25627	0.0087	0.31481	0.0069	0.22059	0.0051	0.2436
2200	0.1045	0.1637	0.0199	0.25604	0.0092	0.29862	0.0067	0.22442	0.004	0.23796
2300	0.0828	0.16615	0.0167	0.25729	0.0074	0.30577	0.0046	0.22915	0.0033	0.23298
2400	0.0897	0.16793	0.0207	0.25769	0.0095	0.30576	0.005	0.22652	0.004	0.23605

TABLE I: Training Loss and Pearson Cosine Similarity across Layers and Steps with Legal Bert Base

	Top 1 Layer		Top 2 Layers		Top 3 Layers	
Step	Training Loss	Pearson Cosine	Training Loss	Pearson Cosine	Training Loss	Pearson Cosine
100	0.2883	0.17629	0.2811	0.14805	0.2815	0.15615
200	0.2611	0.17317	0.271	0.23909	0.2679	0.23149
300	0.2466	0.20133	0.2353	0.22791	0.2363	0.23118
400	0.2271	0.19951	0.2067	0.2652	0.2083	0.25315
500	0.2319	0.21113	0.2077	0.23101	0.2138	0.2331
600	0.1969	0.20473	0.1413	0.24139	0.1543	0.25074
700	0.2024	0.22169	0.1614	0.28842	0.1718	0.26049
800	0.1886	0.2025	0.1119	0.18808	0.134	0.2027
900	0.1669	0.22157	0.1067	0.25754	0.1148	0.25018
1000	0.1744	0.20775	0.0859	0.22454	0.1082	0.21942
1100	0.1424	0.21856	0.0622	0.22065	0.081	0.21636
1200	0.1479	0.20496	0.0694	0.23062	0.0853	0.2485
1300	0.122	0.20422	0.0421	0.21853	0.0595	0.23627
1400	0.1314	0.21429	0.0442	0.22678	0.0633	0.24243
1500	0.106	0.19839	0.0348	0.22031	0.045	0.20964
1600	0.1069	0.16713	0.0253	0.1945	0.0447	0.26831
1700	0.103	0.18042	0.0292	0.21028	0.0426	0.21801
1800	0.0871	0.16774	0.019	0.18985	0.0288	0.20411
1900	0.0885	0.15344	0.0199	0.16654	0.0329	0.20724
2000	0.0834	0.16892	0.0142	0.18741	0.0239	0.21528
2100	0.0715	0.17487	0.0139	0.17835	0.0231	0.22947
2200	0.0756	0.15864	0.0112	0.18398	0.023	0.21725
2300	0.0634	0.16178	0.0094	0.18869	0.017	0.22982
2400	0.0642	0.16664	0.0097	0.19595	0.019	0.21704
2500	0.0573	0.15298	0.0075	0.21271	0.013	0.20405
2600	0.0548	0.15764	0.0076	0.19903	0.0143	0.22342
2700	0.0541	0.15486	0.0069	0.19781	0.0127	0.20778
2800	0.0541	0.14716	0.0061	0.20512	0.0122	0.22931
2900	0.0527	0.13554	0.0059	0.20246	0.0144	0.23629
3000	0.0471	0.13362	0.0044	0.19527	0.0094	0.23564
3100	0.0509	0.14823	0.0046	0.2003	0.0094	0.21759
3200	0.0531	0.14731	0.0051	0.19897	0.0087	0.215
3300	0.0421	0.14571	0.004	0.20924	0.0085	0.2244
3400	0.0489	0.14761	0.0038	0.20484	0.0093	0.21635
3500	0.0435	0.1467	0.0033	0.20698	0.008	0.22047
3600	0.0453	0.14716	0.0037	0.20942	0.0072	0.2221

TABLE II: Training Loss and Pearson Cosine Similarity across Layers and Steps with Legal Bert Small

Frozen Layers	**Epoch**	1	2	3	4	5	6	7	8	9	10
5	Train Loss	0.7004	0.6900	0.6697	0.6486	0.6275	0.6046	0.5667	0.5213	0.4909	0.4362
	Val Loss	0.6921	0.6933	0.6847	0.7318	0.7615	0.7439	0.7491	0.7737	0.8399	0.8688
4	Train Loss	0.7037	0.6921	0.6832	0.6493	0.6214	0.5428	0.4846	0.4270	0.3247	0.2721
	Val Loss	0.6899	0.6865	0.6807	0.6821	0.6992	0.7541	0.8241	0.9407	1.0408	1.1630
3	Train Loss	0.7073	0.6863	0.6680	0.6339	0.5564	0.4530	0.3631	0.2812	0.1856	0.1594
	Val Loss	0.6805	0.7225	0.6718	0.6671	0.7647	0.8315	0.8378	1.0029	1.2208	1.3190

TABLE III: Training Loss and Validation Loss across Epochs with Legal Bert Small Classification-Based Fine-Tuning

loss and Pearson cosine similarity scores at each stage, are summarized in Table II.

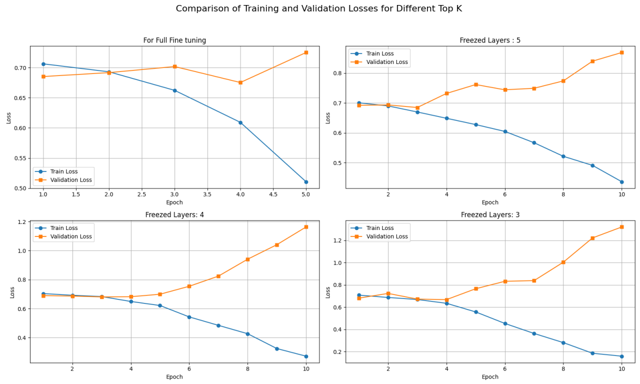


Fig. 6: Classification-Based Fine-Tuning of Legal-BERT Small Using Top-K Layer Adaptation

However, this embedding-based similarity framework consistently yielded unsatisfactory results. To improve upon the embedding-based similarity approach, a more direct sequence classification framework using the ‘nlpaueb/legal-bert-small-uncased‘ model was opted. This method formulates legal entailment as a binary classification task, where each query-article pair is treated as a premise-hypothesis input to a fine-tuned BERT model. The model uses the standard NLI-style tokenization format with ‘[CLS] query [SEP] article [SEP]‘, along with segment and attention masks to distinguish the two inputs. Unlike the previous method that relied on cosine similarity of embeddings, this approach fine-tunes the layers of the LegalBERT-small model using cross-entropy loss, optimizing directly for the ”Yes”/”No” entailment labels. This classification-based method aligns closely with standard natural language inference pipelines, making it more suitable for structured legal entailment tasks where precise label prediction is critical.

The model was fine-tuned for 10 epochs using the Top-K layer freezing strategy, wherein only the upper K transformer layers were updated during training. The corresponding training and validation loss curves are illustrated in Figure 6, and detailed numerical results are reported in Table III. The training and validation loss trends indicate that while the model was able to steadily minimize training loss across all freezing strategies, the validation loss did not show consistent improvement, suggesting potential overfitting. With 5 frozen layers, the model achieved its lowest validation loss (0.6847) at epoch 3, which was the best among all configurations. However, as training progressed, the validation loss increased for all setups, particularly for 4 and 3 frozen layers, indicating poor generalization. Thus, although freezing 5 layers showed relatively better early-stage performance, none of the configurations achieved satisfactory validation stability or final performance.

IV. Results and Analysis

The effectiveness of the suggested legal textual entailment system was assessed according to its capacity to execute article retrieval (Task 3) and entailment classification (Task 4) as specified by the COLIEE 2025 challenge. The experiments employed the supplied English dataset based on the Japanese Civil Code and RITE data, divided into an 80 percent training set (960 instances) and a 20 percent validation set (240 instances). The system’s foundation depends on the nlpaueb/legal-bert-base-uncased model [1], which is used to create embeddings during the retrieval stage and is fine-tuned for sequence classification in the entailment stage.

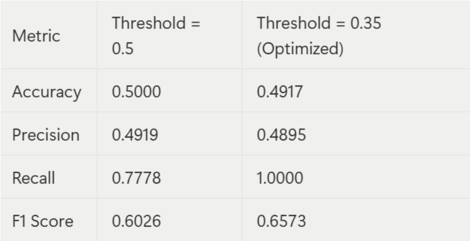
The retrieval component indexed 439 articles from the Japanese Civil Code by creating embeddings with the pre-trained Legal-BERT model. For a specific query, the system calculates its embedding and pulls the top 5 most relevant articles using cosine similarity with the previously computed article embeddings. Although specific retrieval metrics such as Mean Average Precision (MAP) were not directly computed in the notebook, this embedding-based method underpins the selection of candidate articles for the later entailment task.

The entailment model, starting from nlpaueb/legal-bert-base-uncased, was optimized for one epoch on the training data. A layer-freezing approach was utilized, maintaining the first 8 layers of the BERT encoder frozen, leading to around 48.2 percent of the overall parameters being trainable. This method seeks to preserve the overall comprehension of legal language achieved by the lower layers while customizing the upper layers for the particular entailment task. The training procedure employed the AdamW optimizer with a learning rate of 2e-5, resulting in an average training loss of 0.7046 following one epoch.

Assessment on the validation set provided valuable insights into the performance of the entailment model and the effects of threshold optimization. With a standard classification threshold of 0.5, the model attained an accuracy of 0.5000, a precision of 0.4919, a recall of 0.7778, and an F1 score of 0.6026. These metrics show a trend for the model at this threshold to correctly identify a fair amount of true entailment instances (moderate recall) while also erroneously categorizing many non-entailment instances as entailment (low precision), leading to a modest F1 score.

To enhance performance, especially the F1 score that balances precision and recall, a threshold optimization phase was conducted on the validation set. The optimal threshold was found to be 0.35 by assessing F1 scores over various thresholds from 0.3 to 0.8 (in increments of 0.05). Using this refined threshold notably changed the performance attributes. The accuracy decreased slightly to 0.4917, and the precision also fell marginally to 0.4895. The recall surged significantly to 1.0000, indicating the model successfully recognized all genuine entailment instances in the validation set at this threshold. As a result, the F1 score significantly increased to 0.6573. This examination shows a distinct trade-off: the refined threshold enhances the F1 score by emphasizing recall, guaranteeing that likely relevant entailment relationships are identified, though it leads to an increase in false positives. Selecting the ideal threshold is significantly influenced by the particular needs of the application; in a legal setting, prioritizing recall may be favored to prevent overlooking possibly pertinent statutes, even if it necessitates additional manual examination.

The system showcases an effective pipeline for retrieving legal texts and determining entailment through a tailored legal language model. The findings emphasize the efficiency of adjusting pre-trained models such as Legal-BERT and the significance of optimizing thresholds to customize classification performance for particular evaluation metrics, including the F1 score. The validation outcomes, especially the impressive recall obtained with the adjusted threshold after just one training epoch, indicate the model can learn pertinent patterns for the entailment task, but additional training and hyperparameter optimization might further improve precision and overall accuracy.



Metric	Threshold = 0.5	Threshold = 0.35 (Optimized)
Accuracy	0.5000	0.4917
Precision	0.4919	0.4895
Recall	0.7778	1.0000
F1 Score	0.6026	0.6573

Fig. 7: Validation Metrics for Entailment Model.
Note: Metrics calculated on the 20 percent validation set after 1 epoch of training. The optimized threshold (0.35) was selected to maximize the F1 score.

V. Conclusion

The development and fine-tuning of LegalBERT for the legal entailment task yielded promising results, demonstrating the model's capacity to comprehend and reason over complex legal texts. Using sophisticated transformer-based architectures, a binary classification framework was used to address the main goal, which was to ascertain whether a legal hypothesis (query) is logically supported by a law article (context). The main dataset used was the translated Japanese Civil Code, which includes Articles 1 through 724. The data was transformed into query–article–label triplets appropriate for supervised training following XML parsing. Despite the small dataset size of only about 1,200 samples, this organized technique allowed for excellent learning.

To assess the model's performance under various training scenarios, two fine-tuning techniques were used. In order to adequately fine-tune the model to the legal entailment requirement, all 12 layers of LegalBERT had to be updated. Partial fine-tuning, on the other hand, maintained domain-specific generalizations while adapting to the task by freezing the bottom 10 layers and only training the top two, preserving the fundamental legal knowledge inherent in the lower layers. The findings showed that partial fine-tuning provided a good trade-off between task-specific performance and retention of pre-learned legal semantics, especially when trained over a larger number of epochs.

By carefully choosing training strategies and thoroughly evaluating the model, issues including the complexity of legal language, the ambiguity brought about by translation, and the possibility of overfitting because of a short dataset were all properly addressed. Even with limited labeled data, the results show that LegalBERT can be a strong model for legal reasoning tasks with the right fine-tuning and careful architecture considerations. These results validate the viability of implementing transformer-based models in practical legal applications, paving the way for future legal NLP systems that are more resilient and expandable.

VI. Future Recommendations

While the current results are encouraging, several opportunities remain to enhance the performance, scalability, and real-world applicability of the legal entailment model. Addressing limitations such as dataset size, retrieval effectiveness, and domain adaptation will be critical for future development. The following recommendations outline key areas for improvement and exploration:

- Increase and Vary the Dataset: To get around the lack of training data, add more labeled legal datasets from different countries or create artificial cases using data augmentation techniques.
- Improve the Mechanism for Article Retrieval: To increase the relevance of recovered articles for test queries, use sophisticated retrieval approaches as retriever-ranker pipelines or dense vector-based retrieval (e.g., employing Sentence-BERT or LegalBERT embeddings).
- Use Pipelines for Multi-Stage Modeling: Create a two-stage pipeline that applies entailment classification after retrieving the top k pertinent articles. This more closely resembles actual legal search and decision-making.
- Introduce the Modules for Explainability and Justification. To provide legal professionals additional confidence and transparency, incorporate model interpretability characteristics that offer explanations for entailment forecasts.
- Try Out Different Trained Legal Models: To find architecture-specific benefits, investigate and compare more domain-specific language models like CaseLaw-BERT, or more recent transformer models like LegalLLaMA.
- Assess on Actual Legal Tasks: To evaluate practical utility and generality, test the system on downstream tasks such as contract clause validation, precedent matching, and legal document inspection.

Integrating AI technologies like LegalBERT into practical workflows will demand both ethical responsibility and technical rigor as the legal industry develops. These improvements will enable the model to transcend scholarly experimentation and turn into a useful tool for legal research, compliance, and decision support systems.

References

[1] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, ”LEGAL-BERT: The Muppets straight out of Law School,” *Findings of the Association for Computational Linguistics: EMNLP 2020*, Nov. 2020, pp. 2898–2904. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.261

[2] Hugging Face, ”nlpaueb/legal-bert-base-uncased,” *Hugging Face Model Hub*. [Online]. Available: https://huggingface.co/nlpaueb/legal-bert-base-uncased. Accessed: Jun. 01, 2025.

[3] Competition on Legal Information Extraction and Entailment (COLIEE), ”COLIEE 2025,” 2025. [Online]. Available: https://coliee.org/. Accessed: Jun. 01, 2025.

[4] I. Androutsopoulos, P. Malakasiotis, and K. V. Chandrinos, ”A survey of paraphrasing and textual entailment methods,” *Journal of Artificial Intelligence Research*, vol. 38, pp. 135–187, 2010.

[5] I. Dagan, O. Glickman, and B. Magnini, ”The PASCAL recognising textual entailment challenge,” in *Proc. Machine Learning Challenges Workshop*, 2006, pp. 177–190.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, ”Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ”BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proc. NAACL*, 2019.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, ”RoBERTa: A Robustly Optimized BERT Pretraining Approach,” *arXiv preprint arXiv:1907.11692*, 2019.

[9] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, ”LEGAL-BERT: The muppets straight out of law school,” *arXiv preprint arXiv:2010.02559*, 2020.

[10] Z. Zhong, C. Ziems, and D. Yu, ”CaseLaw-BERT: A Pretrained Language Model for Legal Case Retrieval and Entailment,” *arXiv preprint arXiv:2104.08671*, 2022.

[11] R. Rabelo and M. Zampieri, ”Overview of the COLIEE 2021 Competition on Legal Information Extraction and Entailment,” 2021. [Online]. Available: https://sites.ualberta.ca/~rabelo/COLIEE2021.pdf

[12] I. Chalkidis, I. Androutsopoulos, and N. Aletras, ”Neural Legal Judgment Prediction in English,” in *Proc. ACL*, 2019.