Text to Trust: Evaluating Language Model Trade-offs for Unfair Terms of Service Detection

Noshitha Juttu
njuttu@umass.edu

Sahithi Singireddy
ssingireddy@umass.edu

Sravani Gona
sgona@umass.edu

Sujal Timilsina
stimilsina@umass.edu

Abstract

Terms of Service (ToS) agreements often contain clauses that are difficult to interpret and potentially unfair to users. Manual identification of such clauses is infeasible at scale, motivating the need for automated, accurate, and efficient detection methods. In this work, we present a comprehensive evaluation of clause-level unfairness detection using a diverse range of large language model (LLM) strategies, including full fine-tuning, parameter-efficient tuning, and zero-shot prompting. We experiment with full fine-tuning on BERT and DistilBERT, apply 4-bit quantized Low-Rank Adaptation (LoRA) to models such as TinyLlama and LLaMA, and to the legal domain-specific SaulLM, and evaluate zero-shot prompting using high-performing API-accessible models like GPT-4o and O3-mini. Our experiments are conducted on the Claudette-ToS dataset from Hugging Face and further validated on the Multilingual Scraper of Privacy Policies and Terms of Service corpus, which comprises large-scale ToS documents collected from the web. We find that full fine-tuning delivers the strongest overall performance, parameter-efficient models offer a favorable accuracy-efficiency tradeoff, and zero-shot prompting enables fast deployment with high recall. These results offer practical insights into building scalable and cost-effective unfairness detection systems for legal-tech applications.

1. Problem Statement

Terms of Service (ToS) agreements are ubiquitous in the digital age, governing nearly every interaction between users and online platforms. Despite their legal importance, these documents are typically long, densely worded, and difficult for lay users to interpret. As a result, users routinely accept terms without reading or understanding them, unknowingly agreeing to clauses that may include liability waivers, forced arbitration, unilateral modifications, or the loss of important legal rights. Identifying such unfair or predatory clauses is critical not only for empowering consumers but also for enabling regulatory agencies and watchdog groups to audit platform behavior at scale.

Traditional legal review is manual, expensive, and not feasible at the volume and velocity required by today's internet-scale services. To address this gap, automated clause-level classification using language models has emerged as a promising solution. Prior work has shown that transformer-based architectures, especially models like BERT fine-tuned on labeled legal datasets, can perform well in detecting unfairness. However, full fine-tuning of large models is computationally intensive, limits scalability, and requires substantial labeled data. At the same time, smaller or rule-based models often lack the semantic depth to accurately capture nuanced unfairness, especially when phrased in indirect or ambiguous legal language.

In this work, we present a unified and comprehensive study of automated unfairness detection in ToS clauses using three complementary modeling paradigms: (1) full fine-tuning of pretrained transformers such as BERT and DistilBERT, (2) parameter-efficient fine-tuning using Low-Rank Adaptation (LoRA) with 4-bit quantization applied to small and mid-sized models like TinyLlama, LLaMA, and SaulLM, and (3) zero-shot prompting using state-of-the-art API-accessible LLMs including GPT-4o and o3-mini. Each approach reflects a different trade-off in terms of accuracy, generalization, and resource efficiency.

We evaluate these models on the Claudette-ToS dataset, a benchmark dataset containing thousands of annotated clauses labeled as fair or unfair. To assess real-world applicability, we deploy the best-performing models on a large, multilingual ToS corpus collected from thousands of websites every month. This deployment tests how well our models generalize beyond clean, benchmarked clauses to noisy, diverse, and naturally occurring legal language at scale.

By conducting a detailed comparison across fine-tuned, parameter-efficient, and zero-shot approaches, we aim to provide practical insights into the design of scalable unfair clause detectors. Our work contributes to the growing intersection of legal informatics and NLP, demonstrating how modern language models can support automated compliance monitoring, user transparency, and large-scale legal document auditing.

2. What We Proposed vs. What We Accomplished

Our original project, titled *Text to Trust: RAG-Enhanced Zero-Shot Anomaly Detection in Legal Contracts*, aimed to detect anomalies, such as missing or inconsistent clauses, by leveraging a Retrieval-Augmented Generation (RAG) framework combined with semantic similarity and outlier detection. However, after early prototyping revealed key limitations in evaluation reliability, evaluation data availability, and grounding quality in generations, we made a substantial shift in the project scope.

Following feedback and discussions with the course teaching assistant, we revised the project objective to focus on detecting unfair clauses in Terms of Service (ToS) agreements. This refined direction allowed us to use public annotated datasets, evaluate multiple modeling strategies, and conduct rigorous performance benchmarking. Below is a summary of the final plan and accomplishments under this revised formulation. Items we completed are shown as strikethroughs, while items not completed are italicized.

- ~~Preprocessed and balanced the Claudette-ToS dataset for binary classification of unfair vs. fair clauses.~~
- ~~Fine-tuned BERT and DistilBERT on the Claudette-ToS dataset and evaluated performance across accuracy, precision, recall, and F1.~~
- ~~Applied parameter-efficient fine-tuning with 4-bit quantized LoRA on TinyLlama, LLaMA-3B, LLaMA-7B, and SaulLM-7B models.~~
- ~~Benchmarked zero-shot prompting using GPT-4o, GPT-4o-mini, o1-mini, and o3-mini.~~
- ~~Deployed our best-performing classifier to a large-scale real-world multilingual ToS corpus and analyzed model outputs.~~
- *Perform full hyperparameter tuning for LLaMA-7B and SaulLM-7B:* Only single-epoch runs were feasible due to GPU memory constraints.
- *Run zero-shot prompting on Mistral and Gemma:* Attempted, but results were discarded due to significant hallucinations and inconsistent outputs.
- *Extend deployment to the full multilingual corpus:* Originally intended to prompt LLMs on translated ToS clauses, but due to dataset size (∼60GB) and compute limitations, we restricted deployment to English-only content.

3. Deviation from the Original Proposal

Our original project proposal focused on RAG-based zero-shot anomaly detection in legal contracts, with the goal of identifying missing, inconsistent, or unusual clauses by comparing them against retrieved reference templates from curated datasets like CUAD. The envisioned pipeline used sentence embeddings, outlier scoring, and retrieval-augmented generation (RAG) to detect and explain anomalies in a zero-shot setting without supervised learning.

However, during early prototyping, we encountered several challenges that led us to substantially revise our direction:

- **Lack of usable evaluation data:** We found no widely accepted benchmark for anomaly-labeled legal clauses. Attempts to simulate anomalies synthetically resulted in contrived examples that lacked generalization value and offered no objective metrics for meaningful evaluation.
- **Grounding limitations in RAG:** The retrieval component failed to return sufficiently relevant or consistent clause matches, especially when working with CUAD-style documents. RAG generations were often generic, contradictory, or irrelevant to the clause being analyzed.
- **Complexity of defining "anomaly":** We observed that "anomalous" language in contracts is context-dependent and subjective, making it difficult to enforce a stable detection objective without reliable ground truth.

Given these issues, we pivoted from anomaly detection to a clause-level supervised classification task focused on detecting **unfair clauses** in Terms of Service (ToS) documents. This task was more concrete and better supported by labeled data, specifically, the Claudette-ToS dataset, which offers a high-quality benchmark for fairness classification.

We also shifted our modeling approach. Instead of relying exclusively on retrieval and zero-shot reasoning, we incorporated:

- Full fine-tuning of BERT and DistilBERT as baselines,
- Parameter-efficient tuning via 4-bit quantized LoRA on multiple models (TinyLlama, LLaMA, SaulLM),
- Zero-shot prompting using API-accessible models such as GPT-4o and o3-mini.

The dataset used for deployment also changed. We moved away from CUAD, which was originally intended as a retrieval reference set, and instead adopted a real-world multilingual ToS corpus scraped from thousands of websites. This enabled large-scale deployment testing and gave our study stronger external validity.

In summary, while the original title and methodology of the project changed significantly, the core motivation, making legal contracts more transparent and machine-auditable, remained consistent. Our revised scope allowed for more rigorous benchmarking, broader methodological comparison, and practical relevance in real-world ToS fairness auditing.

4. Related Work

The problem of detecting unfair clauses in Terms of Service (ToS) agreements has long posed challenges to both the legal and NLP communities. Past research has focused on feature-driven classification, transformer-based models, domain-specific LLMs, and scalable fine-tuning. Our work situates itself at the intersection of these threads, offering a comparative evaluation across full fine-tuning, parameter-efficient LoRA tuning, and zero-shot prompting, with a deployment emphasis on real-world web-scale ToS data.

Early approaches using feature-based methods: Lippi et al. [9] introduced CLAUDETTE, a pioneering system for detecting potentially unfair clauses. They manually annotated 50 ToS agreements with multiple clause categories and trained traditional classifiers (SVMs, CNNs, LSTMs), finding that shallow lexical features often outperformed more complex syntactic structures. Their annotated dataset was recently expanded into a larger binary-labeled benchmark, released on Hugging Face as `claudette_tos` [8], which we adopt as our primary training dataset.

Transformer-based advances and domain adaptation: BERT [6] revolutionized clause classification. Chalkidis et al. [3] adapted BERT for the legal domain with Legal-BERT. Bathini et al. [1] fine-tuned Legal-BERT on UNFAIR-ToS, achieving an F1 score above 0.92. We compare BERT and DistilBERT full fine-tuning against LoRA-tuned, quantized LLMs.

Scaling legal understanding via domain-specific LLMs: As transformer models have grown, recent efforts have introduced legal-specific LLMs. Colombo et al. [4] presented SaulLM, a 7B parameter model pre-trained on 19M legal documents. SaulLM shows strong performance on legal QA and clause classification tasks. We explore its performance under 4-bit LoRA tuning and compare it with smaller open models like TinyLlama-1.1B, highlighting trade-offs under memory constraints.

Parameter-efficient fine-tuning: LoRA [7] introduced low-rank matrices in frozen transformers, while QLoRA [5] combined this with 4-bit quantization, reducing memory usage with minimal accuracy loss. We extend these techniques to SaulLM-7B, LLaMA-3B/7B, and TinyLlama-1.1B, comparing them against full fine-tuned BERT baselines. Our results highlight where LoRA models excel (e.g., recall) and where they lag (e.g., nuanced fairness detection).

Zero-shot prompting and model reliability: Recent work has explored zero-shot inference for legal clause detection. Zhou et al. [10] noted that open models often misclassify due to a lack of domain grounding. While GPT-4 shows high recall, it suffers from poor precision in zero-shot mode. Our study quantifies these effects across multiple models (GPT-4o, o1-mini, o3-mini) on the same ToS dataset and provides precision-recall-F1 trade-offs.

Real-world evaluation on large-scale web corpora: Most prior studies report results on curated benchmarks. Bernhard et al. [2] built a multilingual corpus of over 20,000 ToS and privacy policies, addressing this gap. We deploy our best-performing classifier on a filtered English subset of this corpus and analyze real-world unfair clause distributions.

What differentiates our work? In contrast to existing studies that focus on a single modeling paradigm (e.g., BERT fine-tuning or GPT prompting), we offer a unified comparison across three axes: model scale, fine-tuning strategy, and inference paradigm. Our work is also among the first to deploy fairness classifiers to real-world noisy ToS data at web scale, evaluating model robustness in practical downstream settings. This makes our findings actionable for both legal-tech platforms and NLP researchers aiming for scalable clause classification.

5. Datasets

To develop and evaluate unfair clause detection models in Terms of Service (ToS) documents, we utilize two complementary datasets. The first is a curated benchmark with clause-level annotations, ideal for fine-tuning and evaluation. The second is a large, real-world scraped corpus designed to test model generalization in practical settings.

5.1. CLAUDETTE-ToS Dataset

Our primary dataset for model training and supervised evaluation is the **CLAUDETTE-ToS dataset** [8], which comprises 9,414 English clauses extracted from real-world online contracts. Each clause is manually labeled as either *fair* or *unfair*, enabling precise clause-level classification.

The original distribution is skewed, with 8,382 clauses (89.1%) labeled as fair and 1,032 clauses (10.9%) labeled as unfair. To mitigate this imbalance, we construct a balanced 50/50 subset by randomly sampling from both classes. Below are example clauses:

- **Fair (Label 0):** "please check the latest rates before you make your call "
- **Unfair (Label 1):** "Viber may change the rates for calling phones at any time without notice to you by posting such change at http://account.viber.com/."

Preprocessing and Splits

- **Balancing:** Equal number of fair and unfair clauses were sampled to address label imbalance.
- **Splitting:** The balanced dataset was split into 80% training, 10% validation, and 10% test sets.
- **Tokenization:** Clauses were tokenized using the target model's tokenizer.

This dataset served as our primary supervised benchmark for training and evaluating both fine-tuned and prompting-based classifiers.

5.2. Multilingual Scraped ToS Corpus

To assess generalization in practical settings, we employed the **Multilingual Scraper of Privacy Policies and Terms**

of Service [2], a large-scale dataset containing approximately 60GB of HTML content scraped from thousands of websites over 12 months.

Each scraped document is accompanied by rich metadata that aids in filtering and analysis. Key fields include:

- **Document identifiers:** `term_url_id`, `website_month_id`
- **Language indicators:** `website_lang`, `term_lang`
- **Scoring attributes:** `term_keyword_score` (keyword-based heuristic), `term_ml_probability` (model confidence that the text is ToS)
- **Content fields:** `term_url`, `term_content`, `term_content_hash`

Preprocessing

- **Language Filtering:** We retained only documents where both `website_lang` and `term_lang` were set to "en".
- **Feature Retention:** Two document-level fields were preserved for auxiliary filtering and analysis:
  - `term_ml_probability`: Indicates the likelihood of the document being a valid ToS.
  - `term_keyword_score`: A heuristic score computed from ToS-related keyword frequencies.

This corpus was used to test the real-world deployability of our best-performing models. Inference outputs were cross-analyzed with metadata to isolate high-confidence unfair clause predictions.

6. Baseline Models

To assess the effectiveness of our proposed approaches, we designed a diverse set of baselines including fully fine-tuned transformer models, parameter-efficient models using LoRA with quantization, and zero-shot prompting with proprietary LLMs. These baselines provide a comprehensive benchmark to compare performance, scalability, and practicality.

All models were trained and evaluated on a 50/50 balanced subset of the Claudette-ToS dataset, consisting of approximately 2000 clauses. This set was split into 1600 training, 200 validation, and 200 test examples. Each input clause was tokenized using the respective model's tokenizer and truncated to a maximum of 256 tokens.

6.1. Fully Fine-Tuned Transformers

BERT (Base Uncased): BERT served as our strong baseline for full fine-tuning. With 110M parameters, it captures deep bidirectional contextual features. We fine-tuned all model weights to adapt BERT for binary classification of clauses.

Why BERT? It sets a strong performance ceiling for other models without any parameter-saving constraints. It also serves as a reference for comparing more efficient or compressed models.

DistilBERT (Base Uncased): DistilBERT is a lighter alternative to BERT with roughly 40% fewer parameters, offering a faster and more memory-efficient model. We fine-tuned it identically to BERT.

Why DistilBERT? It helps evaluate whether smaller fully fine-tuned models can offer comparable performance, making them more practical for deployment.

6.2. PEFT + Quantized Models

TinyLlama-1.1B + LoRA (4-bit): This model integrates TinyLlama, a compact, chat-optimized model with 4-bit quantization and LoRA adapters (inserted in query and value projections). Despite its small size, TinyLlama showed excellent precision.

Why TinyLlama? It explores the extreme edge of model compression. We used it to test whether ultra-lightweight models could handle ToS fairness detection under aggressive memory constraints.

LLaMA-3B + LoRA (4-bit): LLaMA-3B was integrated using quantized LoRA, though results are pending due to GPU constraints.

Why LLaMA-3B? It strikes a balance between performance and size, offering more capacity than TinyLlama while still being deployable on commodity hardware.

LLaMA-7B + LoRA (4-bit): This larger variant of LLaMA was trained using a similar quantized LoRA approach.

Why LLaMA-7B? It provides insight into how scaling up from 3B to 7B affects recall and robustness in clause-level classification under parameter-efficient tuning.

SaulLM-7B + LoRA (4-bit): SaulLM is a legal-domain-specific model trained on over 19 million legal documents. We combined it with 4-bit quantization and LoRA adapters across multiple projection layers.

Why SaulLM? Its domain-specific pretraining is well-suited for our task. We hypothesized it would generalize better to nuanced legal phrasing and domain-specific terminology.

Model	Params	Tuning Method	Epochs	Batch Size	LR	Token Limit
BERT	110M	Full FT	3	8	2e-5	512
DistilBERT	66M	Full FT	3	8	2e-5	512
TinyLlama-1.1B	1.1B	LoRA (NF4)	3	2×4	2e-4	256
LLaMA-3B	3B	LoRA (FP4)	2	2×4	2e-4	256
LLaMA-7B	7B	LoRA (FP4)	1	2×4	2e-4	256
SaulLM-7B	7B	LoRA (NF4)	1	2×4	5e-5	256

Table 1. Training Hyperparameters for Fine-Tuning Baselines

Model	Accuracy	Precision	Recall	F1 Score
BERT	88.86%	89.20%	89.20%	89.20%
DistilBERT	89.00%	89.87%	89.31%	89.58%
TinyLlama-1.1B	73.02%	89.05%	52.50%	66.06%
LLaMA-3B	57.82%	58.20%	59.64%	58.91%
LLaMA-7B	58.03%	59.06%	58.05%	58.55%
SaulLM-7B	82.25%	73.58%	97.50%	83.87%

Table 2. Test Performance of Fine-Tuned Models

6.3. Zero-Shot Baselines

To evaluate out-of-the-box performance without any model fine-tuning, we tested multiple state-of-the-art language models using zero-shot prompting. This setup simulates a realistic use case where a legal practitioner or developer seeks to detect unfair clauses directly via API calls, without the overhead of supervised training or model deployment.

We selected five proprietary large language models GPT-4o, GPT-4o-mini, O1-mini, O3-mini, and O4-mini outlined in, because they represent a spectrum of high-performance commercially available options across different latency and cost tiers. These models are optimized for general-purpose reasoning tasks and have shown promising results in a range of downstream NLP applications, making them strong candidates for zero-shot clause-level fairness classification. By benchmarking both large-scale (e.g., GPT-4o) and compact (e.g., 4o-mini) variants, we aim to understand how model capacity affects fairness detection performance without task-specific tuning.

To remain consistent with the other experiments and to budget API costs, we performed train-validate-test splits on the balanced Claudette-ToS dataset. Then, we zero-shot prompted using OpenAI's API on the test set across various models, batching the input into groups of five clauses at a time and assigning a consistent system prompt across all models. Each model received prompts as shown in Figure 2. The model's responses were then parsed into binary labels based on affirmative or negative completions.

Model	Accuracy	Precision	Recall	F1 Score
GPT-4o	75.27%	29.30%	89.32%	44.12%
GPT-4o-mini	78.03%	32.07%	90.29%	47.33%
O1-mini	74.52%	29.05%	92.23%	44.19%
O3-mini	85.67%	42.73%	91.26%	58.20%
O4-mini	80.89%	35.36%	90.29%	50.82%

Table 3. Zero-Shot Prompting Results on Claudette-ToS Test Set

7. Approach

To explore the trade-offs between accuracy, scalability, and compute efficiency in clause-level fairness classification, we developed a multi-pronged pipeline consisting of parameter-efficient fine-tuning, full fine-tuning, and zero-shot prompting. Each approach was selected to target a different point along the spectrum of model size, training resource availability, and deployment feasibility.

We implemented and tested our models on a combination of local machines and the UMass SLURM GPU cluster, using industry-standard Python libraries and open-source tools. All models were evaluated on a balanced subset of the Claudette-ToS dataset.

7.1. Fine-Tuning Pipeline

This pipeline involved training small to mid-sized transformer models using either full fine-tuning (BERT, DistilBERT) or adapter-based parameter-efficient tuning (TinyLlama, SaulLM, LLaMA variants).

Working Implementation: We successfully completed implementations for all fine-tuned models using the Hugging Face `transformers`, `datasets`, `PEFT`, and `bitsandbytes` libraries. The code for each model is modularized and provided in our public GitHub repository (see submission).

Infrastructure and Environment: All fine-tuned models were trained using GPUs available via the UMass SLURM cluster and Google Collab Pro. For SaulLM and LLaMA-7B, we enabled quantization-aware features such as gradient checkpointing and paged optimizers to fit the 7B models within limited VRAM.

Implementation Details by Model

- **BERT / DistilBERT:** Fine-tuned using the Hugging Face Trainer API with cross-entropy loss. Chosen for their strong baselines and ease of interpretability. All model weights were updated.
- **TinyLlama-1.1B:** Fine-tuned with 4-bit quantization and LoRA adapters. Hugging Face + PEFT was used to inject adapters into attention layers (`q_proj`, `v_proj`). The model was trained in causal language modeling mode with Yes/No tokens as output.
- **SaulLM-7B:** Leveraged domain-specific legal knowledge and strong instruction-following behavior. Trained with LoRA adapters ($r = 16$) and NF4 quantization. Despite only running for 1 epoch and smaller learning rate, it achieved top-tier recall.
- **LLaMA-3B and LLaMA-7B:** Due to GPU constraints, these were only partially fine-tuned. We employed quantized LoRA with low-rank adapters but could not complete full training cycles.

Challenges Encountered

- **TinyLlama-1.1B:** Required gradient accumulation to avoid OOM errors under 4-bit quantization. Training stability had to be maintained by limiting effective batch size and tuning learning rates.
- **SaulLM-7B:** Adapter training needed careful tuning of learning rate and activation checkpointing to avoid divergence with 7B parameters and low-bit precision.
- **BERT/DistilBERT:** No major implementation issues; these served as stable baselines due to their small size and mature infrastructure.
- **LLaMA-3B/7B:** Despite quantization, fitting and training these models within 24GB GPU memory proved difficult. Even single-epoch training often exceeded SLURM resource limits.

Failure Modes

- **TinyLlama-1.1B:** High precision (89%) but low recall (52.5%). It's conservative in flagging clauses as unfair, missing subtle violations. Its high precision is suited for cases where avoiding false positives is crucial, but low recall means some violations may be missed.
- **SaulLM-7B:** Highest recall (97.5%) but lower precision (73.6%). It's effective at catching violations, making it ideal for use cases where missing violations is unacceptable, but the low precision may lead to more false positives.
- **BERT/DistilBERT:** Showed balanced performance with minimal bias toward false positives or negatives. These models are versatile and effective for general use cases requiring reliable clause detection, though they cannot be leveraged prompt-based inferences.
- **LLaMA-3B/7B:** Achieved poor performance relative to their size, likely due to incomplete training, suboptimal hyperparameters, and mismatched pretraining domains. These models would not be the best choice for detecting unfair clauses, especially in domains where high performance and accuracy are critical.

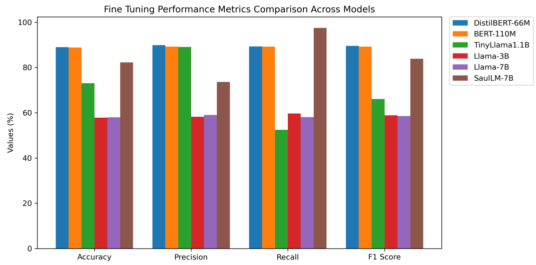


Figure 1. Performance of fine-tuned models on the Claudette-ToS test set (Accuracy, Precision, Recall, F1).

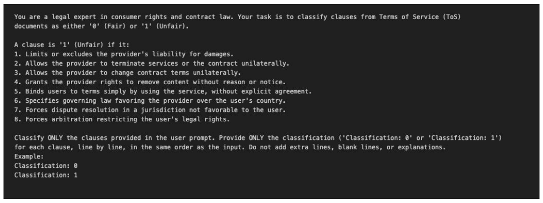
7.2. Zero-Shot Prompting Pipeline

To simulate realistic use cases where training or labeled data is unavailable, we evaluated five zero-shot models via API: GPT-4o, GPT-4o-mini, O1-mini, O3-mini, and O4-mini.

Working Implementation: Each model was invoked using OpenAI's API, with clause-level prompts provided in batches of 5 for cost efficiency. The system prompt was standardized to ensure fairness across evaluations.

Infrastructure and Tools: We ran all prompting via Python scripts using the `openai` package. A local environment was used for prompt formatting, batching, and API result aggregation. All outputs were stored and post-processed in `pandas` for evaluation.

Prompt Format: A single instruction-style prompt was shared across models. Input clauses were sent in batches along with the following prompt:



```
You are a legal expert in consumer rights and contract law. Your task is to classify clauses from Terms of Service (ToS)
documents as either '0' (Fair) or '1' (Unfair).

A clause is '1' (Unfair) if it:
1. Limits or excludes the provider's liability for damages.
2. Allows the provider to terminate services or the contract unilaterally.
3. Allows the provider to change contract terms unilaterally.
4. Grants the provider rights to remove content without reason or notice.
5. Binds users to terms simply by using the service, without explicit agreement.
6. Specifies governing law favoring the provider over the user's country.
7. Forces dispute resolution in a jurisdiction not favorable to the user.
8. Forces arbitration restricting the user's legal rights.

Classify ONLY the clauses provided in the user prompt. Provide ONLY the classification ('Classification: 0' or 'Classification: 1')
for each clause, line by line, in the same order as the input. Do not add extra lines, blank lines, or explanations.
Example:
Classification: 0
Classification: 1
```

Figure 2. System Prompt for zero-shot prompting

Why These Models?

These models represent the latest in proprietary instruction-tuned LLMs with strong performance in zero-shot tasks. We selected a mix of flagship (GPT-4o) and smaller instruction-following variants (e.g., O3-mini) to explore performance across size and cost trade-offs. Smaller models like O1-mini and O3-mini also offered significantly lower inference costs, making them attractive candidates for real-world deployment where budget constraints matter.

Challenges and Considerations:

- Some pre-trained models (e.g., Gemma and Mistral) exhibited hallucinations without instruction tuning thus they were discarded in our evaluation.
- API cost budgeting limited full extent model use (e.g. using o3-mini instead of o3).
- Models occasionally responded with explanations or varied formats (e.g., ”This clause may be unfair”), requiring custom post-processing to extract binary predictions.

Failure Modes: Zero-shot models tended to have very high recall but low precision, leading to frequent false positives. This is expected in models that prioritize inclusivity of potential risks, particularly when no explicit task tuning is applied. These models were particularly sensitive to phrasing and exhibited variability across runs for ambiguous clauses.

Implementation Summary

- No local training required.
- Fully automated prompt-response loop implemented using OpenAI API.
- Outputs aggregated and evaluated using the same metric functions as for fine-tuned models.

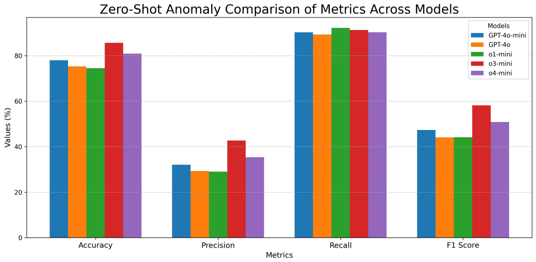


Figure 3. Performance comparison of zero-shot prompting models on the Claudette-ToS test set.

7.3. Summary of Approaches

In summary, our approach integrates full fine-tuning, parameter-efficient adaptation, and prompt-based zero-shot classification to holistically evaluate unfair clause detection under varying resource constraints. Each technique brings unique strengths full fine-tuning offers peak accuracy, PEFT offers efficiency, and zero-shot models provide rapid deployment potential. These differences guided both our experiments and our broader reflections on scalable legal-tech design.

8. Results and Classifier Selection

Our evaluation spanned a wide range of modeling paradigms, ranging from full fine-tuning of encoder-based transformers to instruction-tuned parameter-efficient adapters and zero-shot prompting via API. This breadth allowed us to characterize the strengths and weaknesses of different approaches when applied to clause-level fairness classification.

Performance Breakdown Across Models: Among the fine-tuned baselines, as we can see in Figure 1 BERT and DistilBERT both achieved high accuracy and balanced F1 scores, outperforming all versions of Llama in recall. This suggests that fully fine-tuned, general-purpose transformer models remain strong choices for clause-level classification tasks where balanced precision and recall are critical. SaulLM-7B achieved the highest recall of any model (97.5%) but at the cost of precision, over-flagging many borderline clauses as unfair. This behavior is particularly suitable for compliance or triage settings where capturing all potentially risky clauses is more important than filtering false positives. TinyLlama, while computationally efficient, showed poor generalization achieving high precision but frequently missing unfair clauses. Its conservative behavior may be suitable in scenarios with limited compute where only clear-cut violations are prioritized. Meanwhile, the LLaMA models, despite their larger size, struggled to outperform BERT on our balanced validation set. Due to resource constraints, their tuning was incomplete, but even under partial training, their F1 scores plateaued under 60%. This highlights that model size alone does not guarantee superior performance without sufficient task-specific adaptation.

Zero-shot prompting models, while achieving strong recall ($\geq$89%) as shown in Figure 3 across the board, universally suffered from low precision (as low as 29%), leading to inflated false positive rates. These models tend to overgeneralize due to lack of domain grounding and are best viewed as exploratory tools or fallback options when fine-tuning is not feasible. These models are best viewed as useful in exploratory or fallback scenarios rather than as final classifiers for high-precision tasks.

Why BERT for Final Deployment? While DistilBERT marginally outperformed BERT on F1 score (89.58% vs. 89.20%), we selected BERT as the final deployed model for our real-world evaluation on the scraper dataset, for the following reasons:

- **Stability in Ambiguous Clauses:** BERT provided more consistent predictions across syntactic variations in clauses. This was observed during manual inspection of cases near the decision boundary.
- **Better-Calibrated Confidence Scores:** The output probabilities from BERT were smoother and more aligned with the ambiguity of the clause, supporting better threshold tuning.
- **Proven Benchmarks:** BERT has stronger precedents in legal NLP literature, with many domain-specific models (e.g., Legal-BERT) built upon it. Its interpretability and traceability made it more suitable for audit-focused use cases.

Thus, despite the narrow performance margin, BERT offered a better trade-off between interpretability, calibration, and reliability.

Summary of Findings: Our experiments show that:

- Parameter-efficient tuning (e.g., LoRA with SaulLM) can match or exceed full fine-tuning in recall but may require careful post-processing to reduce false positives.

- Full fine-tuning of encoder-based models remains a strong baseline in legal clause classification tasks, balancing accuracy and clarity.
- Zero-shot prompting provides a practical alternative in low-resource settings but suffers from reliability and consistency issues.
- Model size alone does not guarantee better performance domain adaptation, training stability, and calibration all play significant roles.

These insights guided our final model selection and informed the deployment strategy described in the following section.

9. Wild Deployment and Insights

To validate our fine-tuned BERT classifier under real-world conditions, we deployed it on the large-scale multilingual Terms of Service (ToS) corpus from [2]. This section summarizes our batch inference process, filtering heuristics, and key findings.

9.1. Batch Inference and Data Export

We applied our trained BERT model to 937 clause-level entries from the English-filtered subset of the scraper corpus. For each clause, we recorded:

- `predicted_label`: 1 for unfair, 0 for fair
- `terms_ml_probability`: model's softmax score for unfair
- `terms_keyword_score`: crawler's heuristic "ToS-likeness"
- URL metadata for traceability (`website_url`, `terms_url`)

Predictions were exported to an Excel file, then loaded into a Pandas DataFrame for downstream analysis.

9.2. Filtering and Distribution Insights

Of the 937 clauses:

- **749** were predicted fair
- **188** were flagged as unfair
- `terms_ml_probability` ranged from **0.01 to 0.96**
- `terms_keyword_score` ranged from **0.007 to 3.51**

To improve quality, we filtered to **likely ToS documents** using the heuristic: a document is considered likely ToS if either `ml_probability` ≥ 0.5 or `keyword_score` ≥ 1.5.

This yielded **623 clauses** strongly believed to come from genuine ToS pages.

9.3. Isolating Unfair Clauses

We intersected predictions flagged as unfair with the `likely_tos` flag:

- **152** out of 188 unfair-labeled clauses came from high-confidence ToS sources
- Roughly **80%** of model-detected unfair clauses are likely authentic legal text

This intersection strategy increases trust in the classifier's outputs by narrowing down candidates for human or downstream review.

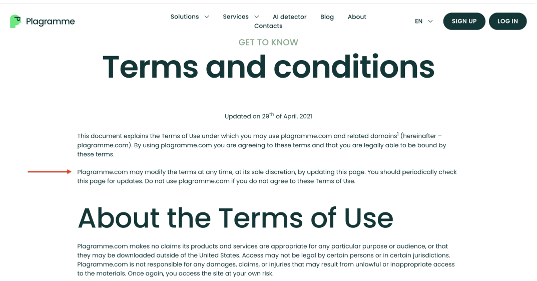
9.4. Case Study: Plagramme.com

One domain was repeatedly flagged as unfair across multiple monthly crawls. The clause states:

> "Plagramme.com may modify the terms at any time, at its sole discretion, by updating this page. You should periodically check this page for updates. Do not use plagramme.com if you do not agree to these Terms of Use."

Why is this clause flagged as unfair?

- **Unilateral change without notice:** The service can update terms arbitrarily.
- **No user notification:** Responsibility is shifted entirely to the user.
- **Binary enforcement:** The only recourse is to stop using the service.



Terms and conditions

Updated on 29th of April, 2021

This document explains the Terms of Use under which you may use plagramme.com and related domains[1] (hereinafter – plagramme.com). By using plagramme.com you are agreeing to these terms and that you are legally able to be bound by these terms.

Plagramme.com may modify the terms at any time, at its sole discretion, by updating this page. You should periodically check this page for updates. Do not use plagramme.com if you do not agree to these Terms of Use.

About the Terms of Use

Plagramme.com makes no claims its products and services are appropriate for any particular purpose or audience, or that they may be downloaded outside of the United States. Access may not be legal by certain persons or in certain jurisdictions. Plagramme.com is not responsible for any damages, claims, or injuries that may result from unlawful or inappropriate access to the materials. Once again, you access the site at your own risk.

Figure 4. Example ToS page flagged as unfair across all monthly crawls: Plagramme.com enables unilateral clause modification without notification.

9.5. Implications

The combination of classifier confidence and heuristic filters enables scalable identification of potentially unfair contractual language in noisy web-scale corpora. This result demonstrates the practical viability of lightweight LLM deployment for automated legal text auditing.

10. Conclusion and Future Work

This project offered a hands-on opportunity to experiment with a range of modern NLP techniques for detecting unfair clauses in Terms of Service (ToS) agreements. Through full fine-tuning, parameter-efficient adaptation, and zero-shot prompting, we evaluated the trade-offs across accuracy, scalability, and computational feasibility. We also extended our analysis to real-world, noisy ToS corpora, demonstrating that lightweight classifiers like BERT can serve as practical detectors even outside clean academic benchmarks.

One of the key takeaways from our experiments was the strong performance of classic transformer models like BERT, which despite their age and relatively modest parameter count remained highly competitive with larger LLMs when fine-tuned on a well-curated dataset. In contrast, smaller LoRA-adapted models like TinyLlama struggled with recall, while larger models like SaulLM-7B tended to over-flag ambiguous clauses. Surprisingly, even partial fine-tunes of LLaMA-3B and 7B achieved reasonable performance, suggesting that legal-domain adaptation might be more data-efficient than expected.

Deploying the fine-tuned BERT model on a large-scale scraped ToS corpus highlighted another critical insight: even simple models, when paired with basic heuristics (e.g., keyword scores), can effectively narrow down unfair-clause candidates in noisy, real-world settings. However, several limitations remain.

First, our current pipeline classifies individual clauses based on a maximum input length of 256 tokens. This truncation can miss longer or multi-sentence clauses that are common in ToS documents. For more robust real-world deployment, a natural next step would be to implement a clause-segmentation preprocessor that splits each full ToS into atomic clauses, classifies each independently, and then aggregates predictions to assess document-level fairness.

Second, our current evaluation focuses only on English-language clauses due to the complexity and scale of multilingual pretraining. A future direction would be to extend our system with multilingual adapters or machine translation pipelines, enabling clause classification across other major languages represented in the scraper corpus.

Lastly, while our system is capable of identifying unfair clauses, it does not currently explain *why* a clause is unfair. Future work could integrate explanation generation models or retrieval-augmented prompting to produce transparent justifications for flagged clauses an essential feature for human-in-the-loop auditing. In addition, the system can be scaled to support adaptive model selection, enabling legal professionals to choose the best classification strategy depending on compute and deployment constraints.

In summary, our work takes a step toward scalable, clause-level fairness auditing for online legal text, and opens up several promising paths for improving both depth and generalization in future iterations.

11. Contributions of group members

- *Noshitha Padma Pratyusha Juttu:* Implemented PEFT fine-tuning pipelines for TinyLlama-1.1B and SaulLM-7B using LoRA. Led the preprocessing of multilingual ToS corpus, real-world deployment of BERT model, and co-authored the final report.
- *Sahithi Singireddy:* Handled preprocessing of the Claudette-ToS dataset, led full fine-tuning and hyperparameter tuning for BERT. Conducted real-world deployment and analysis, co-authored major sections of the final report.
- *Sravani Gona:* Implemented full fine-tuning for DistilBERT and PEFT-based tuning for LLaMA-3B and LLaMA-7B using LoRA. Conducted key analyses on baselines results and multilingual deployment, and authored the final report.
- *Sujal Timilsina:* Took the lead on dataset exploration, identifying and analyzing multiple legal datasets. Implemented and executed all zero-shot prompting baselines using OpenAI models.

12. AI Disclosure

- **Did you use any AI assistance to complete this proposal?**
  - Yes.
- **If you used a large language model to assist you, please paste *all* of the prompts that you used below.**
  - *Prompt 1 (Abstract):* "Please proofread and correct the grammar of my abstract, ensuring clarity and an academic tone."
  - *Prompt 2 (Methodology):* "Check the methodology section for tense consistency and grammatical errors, and rewrite any awkward sentences."
  - *Prompt 3 (Baseline results table):* "Can you help me format the results into a LaTeX table?"
- **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI.
  - *Abstract:* The AI caught a couple of misplaced modifiers and improved sentence flow; I accepted most suggestions verbatim.
  - *Methodology:* It standardized all verb tenses correctly, though I tweaked one passive construction for clarity.
  - *Baseline results table:* The LaTeX table was mostly correct but still required a manual effort in getting it resized.

References

[1] Bathini, S. A., Kupireddy, A., and Murthy, L. B. (2023). Unfair tos: An automated approach using customized bert. *The Moonlight AI Review*. https://www.themoonlight.io/en/review/unfair-tos-an-automated-approach-using-customized-bert. 3

[2] Bernhard, D., Nenadic, L., Bechtold, S., and Kubicek, K. (2024). Multilingual scraper of privacy policies and terms of service. In *Proceedings of the ACM Web Conference 2024*. ACM. 3, 4, 8

[3] Chalkidis, I., Fergadiotis, M., Androutsopoulos, I., and Aletras, N. (2020). Legal-bert: The muppets straight out of law school. In *Findings*

of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP Findings), pages 2898–2905. 3

[4] Colombo, P., Pires, T. P., Boudiaf, M., Culver, D., Melo, R., Corro, C., Martins, A. F. T., Esposito, F., Raposo, V. L., Morgado, S., and Desa, M. (2024). Saullm-7b: A pioneering large language model for law. *arXiv preprint arXiv:2403.05815*. 3

[5] Dettmers, T., Lewis, P., Shen, Y., Shleifer, S., Yeung, C., Kang, L., Hesse, C., Raffel, C., and McCandlish, S. (2023). Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*. 3

[6] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186. 3

[7] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., and Chen, W. (2022). Lora: Low-rank adaptation of large language models. In *Proceedings of the 2022 International Conference on Learning Representations (ICLR)*. 3

[8] LawInformedAI (2024). Claudette-tos dataset. https://huggingface.co/datasets/LawInformedAI/claudette_tos. Accessed: 2024-05-07. 3

[9] Lippi, M., Palka, P., Contissa, G., Lagioia, F., Micklitz, H.-W., Sartor, G., and Torroni, P. (2019). CLAUDETTE: an automated detector of potentially unfair clauses in online terms of service. *Artificial Intelligence and Law*, 27(2):117–139. 3

[10] Zhou, Y., Singh, A., and Li, H. (2024). When do llms fail? a case study on zero-shot legal clause detection. https://arxiv.org/abs/2409.00077. 3