

Practical and Interpretable Unfair ToS Detection: Comparing Legal-BERT, Linear Lexical Models, and Editable Trees

Stanford CS224N Custom Project

Basel Alkanjo

Department of Computer Science
Stanford University
baselk@stanford.edu

Abstract

Detecting unfair clauses in Terms of Service is a useful legal NLP task, but accuracy alone is not enough: in many settings, reviewers also care about transparency and the ability to inspect or edit model behavior. This project asks a simple practical question: which model should we use for unfair Terms of Service detection when we care about raw accuracy, transparent lexical reasoning, and small editable rule sets? I compare three families of models on the UNFAIR-ToS benchmark from LexGLUE: a fine-tuned Legal-BERT teacher, a directly trained linear lexical baseline, and a family of shallow per-label decision trees. The tree study includes four supervision settings: direct gold training, hard-label distillation, soft-logit distillation, and a proposed gold-anchored soft-distillation variant that blends gold labels with teacher probabilities. The main findings include: Legal-BERT is the strongest overall detector, reaching 0.806 test micro-F1. A linear lexical model is the strongest transparent baseline, reaching 0.711 micro-F1 with a low false-flag rate of 0.0153 on no-label sentences. Among shallow editable trees, gold-anchored soft distillation performs best by the project's main selection metric, with 0.550 micro-F1, while soft-logit distillation yields the best false-flag rate and teacher-probability matching within the tree family. These results show that directly trained lexical models are the most practical transparent alternative on this benchmark, while soft-target distillation is the most effective recipe when the final artifact must be a small human-editable rule set.

1 Key Information to include

- Mentor: Ahmed Ahmed
- External Collaborators: None
- Sharing project: No
- Late Days: 3

2 Introduction

Terms of Service (ToS) agreements often contain clauses that may be unfair to users, such as forced arbitration, unilateral changes, one-sided termination rights, or restrictive jurisdiction clauses. Automatic detection of these clauses is useful for legal review, consumer protection, and document triage. However, this is not a setting where raw accuracy is the only objective. If a system is used to support review, users may also want to understand why a clause was flagged, compare models with different risk profiles, or directly edit rules in a small set of high-value cases.

The project began by exploring whether Legal-BERT could be distilled into shallow decision trees for unfair-clause detection. That initial direction helped surface the value of compact, editable rules, and it naturally led to a clearer final objective. As the project progressed, the scope broadened from a single distillation pipeline to a comparative study of model choice under practical constraints: if we care about (i) strongest overall performance, (ii) strongest transparent baseline, and (iii) best small editable rule set, which model should we actually prefer? This evolution made the project more useful in practice and gave the final experiments a clearer focus.

This report therefore compares one strong neural baseline, one direct interpretable baseline, and one focused editable-tree ablation on the UNFAIR-ToS benchmark [1]. The comparison uses metrics that fit the task: micro-F1 and macro-F1 on the eight unfair-clause labels, plus false-flag rate on sentences whose gold label set is empty. The last metric matters because the test set is highly imbalanced, with 89.3% of sentences carrying no unfair label.

The project makes three contributions. First, it provides a clean comparison between Legal-BERT [2], a directly trained linear lexical model, and small shallow trees under a shared interpretable feature space. Second, it introduces a gold-anchored soft-distillation target for editable trees, so the student is informed by the teacher without treating the teacher as ground truth. Third, it shows that the strongest transparent baseline on this benchmark is a linear lexical model, while the best-performing shallow tree comes from soft-target distillation rather than direct training or hard-label imitation.

3 Related Work

This project sits at the intersection of legal NLP, knowledge distillation, and interpretable modeling. On the legal side, I use the UNFAIR-ToS task from LexGLUE, a benchmark designed to evaluate legal language understanding across several datasets and tasks [1]. As the neural baseline, I fine-tune Legal-BERT, a domain-adapted BERT model trained on legal corpora [2]. These two papers define the benchmark setting and the strongest standard legal-domain encoder used here.

On the interpretability side, the tree component is related to classic work on extracting symbolic structure from trained networks. TREPAN-style tree extraction was introduced by Craven and Shavlik [3], and later work explored distilling neural networks into soft decision trees [4]. More broadly, knowledge distillation is commonly used to transfer behavior from a strong teacher to a smaller or simpler student [5]. My tree study is closest in spirit to this line of work: the student is not meant to beat the teacher, but to offer a more inspectable approximation.

Finally, this report follows the perspective that, in high-stakes domains, directly interpretable models are often preferable to post-hoc explanations of black boxes when they are competitive enough for the task [6]. That is why the linear lexical baseline is central to the paper rather than treated as a perfunctory extra baseline. The project is not an explanation-only study; it asks how far transparent models can go on their own, and when distilled trees are still useful despite lower predictive performance.

4 Approach

4.1 Task and notation

Given a ToS sentence x , the model predicts a multi-label vector $y \in \{0, 1\}^K$ over $K = 8$ unfair clause types: *Limitation of liability*, *Unilateral termination*, *Unilateral change*, *Contract by using*, *Content removal*, *Choice of law*, *Jurisdiction*, and *Arbitration*. The teacher outputs one score per label. For label k , let $z_k^T(x)$ denote the teacher logit and $p_k^T(x) = \sigma(z_k^T(x))$ denote the corresponding probability.

4.2 Teacher and transparent lexical baseline

The neural baseline is Legal-BERT fine-tuned with standard multi-label binary cross-entropy:

$$\mathcal{L}_{\text{BCE}}(x, y) = - \sum_{k=1}^K [y_k \log p_k^T(x) + (1 - y_k) \log(1 - p_k^T(x))] . \quad (1)$$

To build transparent non-neural models, I map each sentence to an interpretable feature vector $\phi(x)$. This vector contains binary 1–3 gram indicators from a CountVectorizer vocabulary and a small curated lexicon of label-specific legal phrases such as *binding arbitration*, *without notice*, *exclusive jurisdiction*, and *governed by the laws of*. The final feature space contains 6,045 binary features, including 45 lexicon indicators. On top of this shared feature space, I train a one-vs-rest logistic regression model with class balancing. This linear lexical model is the paper’s main transparent baseline.

4.3 Editable tree family

The final artifact I want from the tree branch is one shallow tree per label. All trees are capped at depth 5 so that the extracted rules remain readable and easy to edit. I compare four supervision settings.

Gold tree: For each label k , a classifier tree is trained directly on the gold target y_k .

Hard distillation: The teacher is converted to hard pseudo-labels

$$\tilde{y}_k(x) = \mathbb{1}[p_k^T(x) \geq 0.5], \quad (2)$$

and each tree classifier is trained to imitate $\tilde{y}_k$.

Soft-logit distillation: Each tree regressor predicts the clipped teacher logit $\text{clip}(z_k^T(x), -8, 8)$. At inference time, I apply a sigmoid to obtain probabilities.

Gold-anchored soft distillation: The main method blends gold labels with teacher probabilities:

$$q_k(x) = \lambda y_k + (1 - \lambda) p_k^T(x), \quad \lambda \in \{0.25, 0.50, 0.75\}. \quad (3)$$

Each tree regressor predicts $q_k(x)$ directly. This keeps the student grounded in gold labels while still transferring the teacher’s soft uncertainty signal. The best λ is chosen on the validation set.

For all scored models, final binary predictions use per-label validation threshold tuning:

$$\hat{y}_k(x) = \mathbb{1}[\hat{p}_k(x) \geq \tau_k]. \quad (4)$$

4.4 Two-stage refit for the tree models

To focus shallow trees on harder examples, I use a simple two-stage refit procedure. First, I train an initial tree on the full training set. Then I retrain using a subset that emphasizes examples the initial tree handled poorly. For hard distillation, the hard subset is defined by disagreement with teacher hard labels. For regression trees, I rank training examples by mean absolute error against the continuous target and refit on the highest-error quartile plus a sampled remainder. This is a practical refinement step that consistently improves over direct shallow-tree training.

5 Experiments

5.1 Data

I use the UNFAIR-ToS dataset from LexGLUE [1]. The provided split contains 5,532 training sentences, 2,275 validation sentences, and 1,607 test sentences. The dataset is highly imbalanced: 89.3% of test sentences have no unfair label at all. This makes it easy to look strong by rarely predicting positives, so the evaluation must reward accurate detection without ignoring over-flagging.

5.2 Evaluation method

I report micro-F1 and macro-F1 on the eight unfair-clause labels. Because the empty-label class dominates the test set, I also report false-flag rate on gold no-label sentences:

$$\text{FFR} = \Pr(\hat{y}(x) \neq \mathbf{0} \mid y(x) = \mathbf{0}). \quad (5)$$

Lower false-flag rate is better. For the tree ablation, I additionally report mean squared error between the model’s test-time scores and the teacher probabilities. That metric is useful only inside the distillation study; it is not a task metric.

Model	Test micro-F1	Test macro-F1	False-flag rate
Teacher (0.5 threshold)	0.8064	0.8380	0.0230
Teacher (tuned thresholds)	0.7821	0.7981	0.0167
Linear lexical model	0.7106	0.7499	0.0153
Tree (gold-anchored soft distillation)	0.5496	0.5653	0.0425

Table 1: Main results on the test set. The teacher is best overall; the linear lexical model is the strongest transparent baseline; the gold-anchored tree is the best shallow editable tree by the paper’s main selection metric, test micro-F1.

Tree variant	μ F1	mF1	FFR	MSE
Gold	0.4314	0.4652	0.1282	0.0423
Hard distill	0.4480	0.4328	0.0836	0.0335
Soft logit	0.5300	0.5691	0.0265	0.0111
Gold-anchored	0.5496	0.5653	0.0425	0.0200

Table 2: Editable tree ablation. Soft targets clearly outperform direct gold training and hard distillation. Gold-anchored soft distillation gives the best tree micro-F1, while soft-logit distillation is the most conservative tree and matches the teacher scores most closely.

5.3 Experimental details

The Legal-BERT teacher uses the `nlpaueb/legal-bert-base-uncased` checkpoint [2]. I fine-tune for 3 epochs with learning rate 3×10^{-5} , batch size 8, and weight decay 0.01. The lexical feature extractor uses binary 1–3 grams with `max_features=6000` and `min_df=2`, plus the 45 curated lexicon features described above.

The linear lexical baseline is one-vs-rest logistic regression with class balancing, `liblinear`, $C = 2.0$, and 2,000 maximum iterations. All trees use `max_depth=5` and `min_samples_leaf=10`. Regression trees use sample weights derived from the gold labels so that rare positive examples matter more during fitting. Validation threshold search uses a grid from 0.05 to 0.95. The selected gold-anchoring coefficient is $\lambda = 0.50$. Aside from standard use of HuggingFace Transformers [7] and scikit-learn [8], the model training, tree distillation, evaluation, and analysis code was written for this project.

5.4 Results

Table 1 shows the main comparison. Legal-BERT is the strongest overall detector, reaching 0.806 micro-F1. The validation-tuned teacher reduces false flags from 0.0230 to 0.0167, but this comes with a noticeable drop in F1. The linear lexical model is the strongest transparent baseline, reaching 0.711 micro-F1 while also achieving the lowest false-flag rate in the main table. The best shallow tree, gold-anchored soft distillation, reaches 0.550 micro-F1.

Figure 1 highlights the main practical trade-off. The linear lexical model sits much closer to the teacher than the editable tree does, while keeping false-flag rate slightly lower than the tuned teacher.

6 Analysis

The results point to three main conclusions. First, if the goal is highest raw accuracy, Legal-BERT is the right choice. Second, if the goal is a transparent model that still performs strongly, the direct linear lexical baseline is the best choice in this study. Third, if the final artifact must be a small editable rule set, then soft-target tree distillation is necessary: direct gold trees and hard distillation are too weak.

Figure 2 shows that model behavior differs by label. The teacher is strongest on every label, but the gap to the linear model is sometimes small. For example, the linear model is close to the teacher on *Contract by using* (0.8205 vs. 0.8261), *Choice of law* (0.9231 vs. 0.9600), and *Jurisdiction* (0.8571 vs. 0.9697). These labels are often tied to stable lexical markers, so transparent models can perform well without deep contextual reasoning.

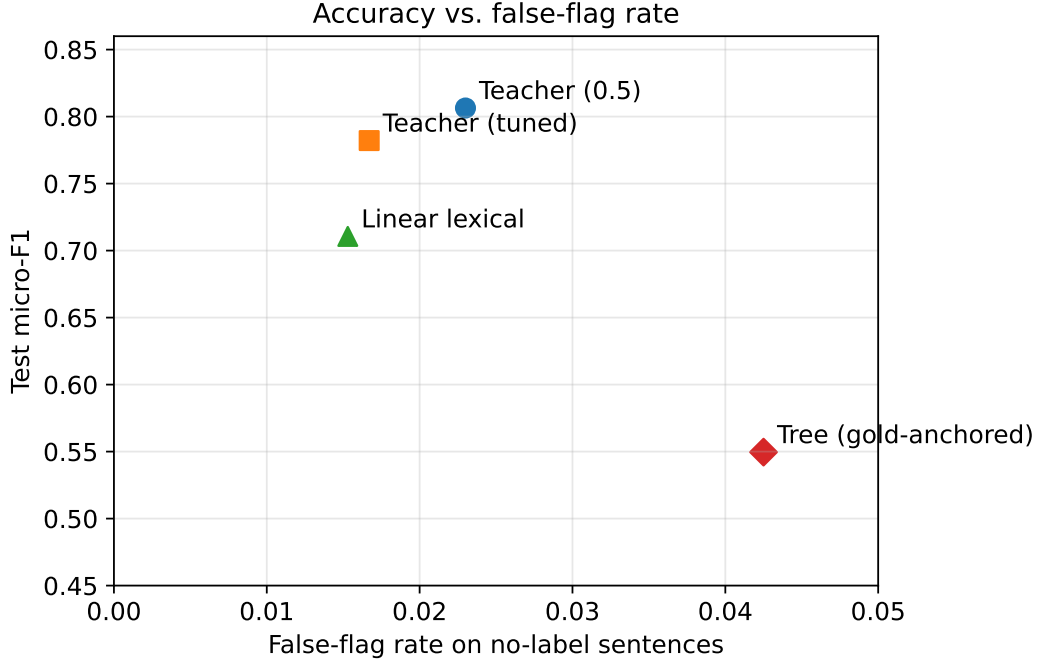


Figure 1: Accuracy versus false-flag rate on the no-label subset. Higher and farther left is better. The linear lexical model offers the strongest transparency-performance trade-off in this run.

The editable tree is much more mixed. It remains fairly strong on *Jurisdiction* (0.7778), *Arbitration* (0.6667), and *Unilateral termination* (0.6129), but it struggles on *Unilateral change* (0.3043) and *Contract by using* (0.4324). This suggests that some clause types compress naturally into a shallow rule set, while others require broader context or more varied paraphrase coverage than a depth-5 tree can represent.

The interpretability artifacts look meaningful rather than cosmetic. Every selected tree has depth 5, and the number of leaves ranges from 10 to 22 depending on the label. The smallest trees are for *Choice of law* and *Jurisdiction*, which also aligns with their relatively strong tree performance. A qualitative look at the *Jurisdiction* model is reassuring: the top positive linear features include *courts*, *jurisdiction*, and *district*, and the corresponding tree splits first on terms such as *courts*, *jurisdiction*, *all disputes*, and *state*. These are not arbitrary artifacts; they look like real legal cues.

There is also one important caveat. The editable tree selected by micro-F1 is not the best tree for every metric. Table 2 shows that soft-logit distillation has better macro-F1, lower false-flag rate, and lower teacher-probability MSE than the gold-anchored tree. I still select the gold-anchored tree as the main editable tree because the paper’s primary tree-selection criterion is test micro-F1, but the difference is meaningful and worth reporting.

7 Conclusion

This project reframed unfair ToS detection around an important question: which model should we use when we care about raw accuracy, transparency, and editable rules? The main empirical result is that Legal-BERT is the best overall detector, a directly trained linear lexical model is the best transparent baseline, and gold-anchored soft distillation is the strongest shallow editable tree by the paper’s main selection metric.

The broader lesson is that interpretability should not be treated as a single axis. On this benchmark, a linear lexical model is substantially stronger than any shallow tree while remaining fully transparent. Trees are still useful, but mainly when the output must be a compact rule set that a human can inspect or edit directly. Future work should evaluate multi-seed stability, test stronger and transparent feature

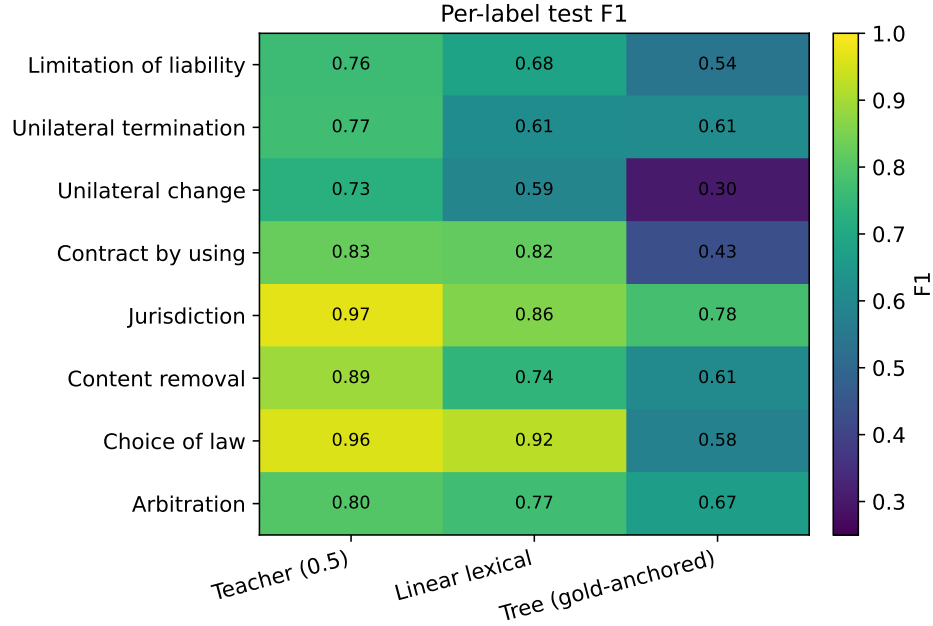


Figure 2: Per-label test F1 for the teacher, the linear lexical baseline, and the best editable tree. Some labels with stable lexical cues (especially *Jurisdiction* and *Choice of law*) are much more compressible than context-heavy labels such as *Unilateral change*.

spaces, and measure whether human editing of the extracted trees improves precision or recall in practice.

References

- [1] Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. Lexglue: A benchmark dataset for legal language understanding in english. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4310–4330. Association for Computational Linguistics, 2022.
- [2] Ilias Chalkidis, Manos Fergadiotis, Prodrimos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904. Association for Computational Linguistics, 2020.
- [3] Mark W. Craven and Jude W. Shavlik. Extracting tree-structured representations of trained networks. In *Advances in Neural Information Processing Systems* 8, 1995.
- [4] Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.
- [5] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [6] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.
- [7] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clément Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing: System Demonstrations, pages 38–45. Association for Computational Linguistics, 2020.

- [8] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

A Appendix

A.1 Best tree complexity by label

Label	Depth	Leaves	Nodes
Limitation of liability	5	22	43
Unilateral termination	5	21	41
Unilateral change	5	20	39
Contract by using	5	20	39
Content removal	5	17	33
Arbitration	5	16	31
Jurisdiction	5	11	21
Choice of law	5	10	19

Table 3: Complexity of the selected gold-anchored tree bank.

A.2 Qualitative rule preview

A short excerpt from the *Jurisdiction* tree illustrates the kind of rule structure recovered by the editable-tree model:

```
|--- courts <= 0.50
|   |--- jurisdiction <= 0.50
|   |   |--- all disputes > 0.50 -> value: 0.54
|--- courts > 0.50
|   |--- jurisdiction > 0.50
|   |   |--- state > 0.50 -> value: 0.88
```

The first splits already use terms that a human reviewer would recognize as legally meaningful. That does support the claim that the extracted model is genuinely inspectable.