

# Automated Unfair Clause Detection in Terms & Services for User Right Protection Using DistilBERT-ERT - Machine Learning- “ClauseNLP”

Rohit Paswan, Arnav Dhiwar, Sumed Hubale

PDEA's College of Engineering Manjri (Bk)

Department of Information Technology

Pune, Maharashtra, India

Savitribai Phule Pune University

Student Email: [paswanrohit2345678@gmail.com](mailto:paswanrohit2345678@gmail.com),

[arnavdhiwar07@gmail.com](mailto:arnavdhiwar07@gmail.com),

[sumedhubale220@gmail.com](mailto:sumedhubale220@gmail.com)

## ABSTRACT

Digital Terms of Service (ToS) agreements have become an unavoidable aspect of online interaction, yet the overwhelming majority of users never read them before clicking 'agree.' These documents are typically dense, lengthy, and laced with legal jargon that even educated readers find difficult to parse. Hidden within such agreements are clauses that may unfairly limit user rights, allow extensive data collection, or waive legal protection entirely. Addressing this growing concern, this paper presents ClauseNLP — an AI-powered web-based system designed to automatically analyze Terms of Service documents and surface potentially harmful clauses in plain, understandable language.

The system leverages DistilBERT, a lightweight transformer model distilled from BERT, fine-tuned on a labeled dataset of ToS clauses categorized into three risk levels: Risky, Moderate, and Safe. Beyond clause classification, the system integrates a local large language model (Ollama/LLaMA) to generate natural language summaries of the analyzed document, offering users a concise risk report without needing to read the full agreement. Supporting both URL-based and PDF-based input, ClauseNLP achieves a classification accuracy of 80.14% and demonstrates practical utility as a consumer-facing transparency tool in the digital ecosystem.

**Keywords** — Terms of Service analysis, DistilBERT, NLP clause classification, unfair clause detection, consumer protection, transformer models, risk scoring,

## 1. INTRODUCTION

The existing web ecosystem functions inside a progressively intricate arrangement of service utilization conditions. Spanning from digital communication networks to cloud-based preservation, nearly each web-based utility demands participant acceptance of predetermined provisions prior to operation. Nevertheless, in reality, these arrangements predominantly advance institutional advantage rather than consumer welfare.

Statistical evidence establishes that less than ten-percent of end-users carefully examine service documentation prior to

agreement. Standard documentation for prominent platforms typically extends past ten thousand words, occasionally matching medium-length novels. Taking into account the agility of web-based innovations, expecting comprehensive participant evaluation proves unrealistic. Implications resulting from unexamined acceptance range across coerced information gathering, elimination of collaborative legal remedies via mandatory settlement clauses, extensive material licensing to suppliers, plus susceptibility to abrupt discontinuation absent safeguards.

Artificial linguistics has undergone considerable development in contemporary periods, furnishing sophisticated mechanisms for comprehending plus systematizing professional documentation. System architectures including BERT alongside comparable successors have exhibited powerful efficiency on discrimination jobs encompassing sophisticated professional terminology. This advancement produces a sensible direction for mechanized service agreement investigation — mechanisms competent at comprehending plus characterizing these materials advantaging ordinary constituencies.

ClauseNLP addresses exactly this purpose. The framework accepts service agreement materials as input — delivered via hyperlink or archive submission — plus supplies an methodical danger assessment coupled with plain-language clarification. The discrimination foundation leverages DistilBERT, adjusted utilizing a curated assemblage of service agreement fragments, whereas articulation generation runs via personally-hosted language system. The product constitutes a functional, confidentiality-protective mechanism for purchaser defense throughout the technology-driven work.

The motivation behind this work stems from a fundamental imbalance in the digital agreement ecosystem. When a user creates an account on any online platform, they are presented with a contract — yet the power to negotiate that contract is entirely absent. The document is a take-it-or-leave-it proposition, crafted by teams of lawyers whose primary objective is to maximize the platform's legal flexibility while minimizing its liability.

Several high-profile incidents have brought this issue to public attention. Platform policy changes that retroactively claimed ownership over user content, widespread data sales to advertisers buried in privacy policies, and the systematic removal of users' right to collective legal action through

buried arbitration clauses — these are not edge cases. They are standard features of most popular online services.

Meanwhile, digital literacy around legal agreements remains low. Even technically sophisticated users often lack legal training to recognize which clauses are standard boilerplate and which represent genuine threats to their interests. There exists a clear need for an automated intermediary — a tool that can bridge the gap between complex legal text and ordinary user comprehension. ClauseNLP was built to fill this gap, using accessible AI technology to restore some measure of informed consent to the online agreement process.

## II. LITERATURE SURVEY

The problem of automated legal document analysis has attracted growing attention from both the NLP and legal informatics communities. Several threads of prior work are directly relevant to this project.

I. The CLAUDETTE project by Lippi et al. [1] introduced the first annotated dataset of unfair clauses drawn from consumer-facing Terms of Service agreements. Using conventional machine learning classifiers, the research demonstrated that automated detection of harmful contractual provisions was both feasible and consistent across different platforms. This foundational work established benchmark performance figures that subsequent research has sought to build upon and remains the most widely cited resource in the field. The study was published in Springer Nature in February 2021.

II. Ruggeri et al. [2] investigated the detection and explanation of unfair clauses in consumer contracts through the use of memory network architectures. Their approach went beyond simple classification by generating explanations alongside predictions, offering a more interpretable output that is directly useful to non-expert users. The research highlighted the value of pairing classification with natural language explanations, a design principle that influenced the summary generation component of ClauseNLP. This work appeared in the Springer journal Artificial Intelligence and Law in March 2022.

III. Vattikuti [3] examined the application of Natural Language Processing to automated legal document analysis and contract review in a study published in the International Journal of Sustainable Development in the Field of IT in December 2024. The paper surveyed a range of NLP techniques applied to contract understanding, emphasizing the practical challenges of processing dense legal language at scale. The work reinforced the importance of clause-level analysis over document-level summarization for achieving actionable and reliable legal insights.

IV. Pal and Rajnala [4] proposed a self-supervised approach to training BERT for legal text classification, published in IEEE Xplore in May 2023. Their method addressed the challenge of limited labeled legal data by leveraging self-supervised pre-training on unlabeled legal corpora before fine-tuning on specific classification tasks. This strategy produced measurable accuracy improvements over standard supervised fine-tuning baselines and demonstrated that transformer models could be effectively adapted for legal NLP even when annotated training data is scarce.

V. Galassi et al. [5] extended the scope of unfair clause detection beyond the English language by developing a multilingual identification system capable of processing ToS documents in several European languages. Published in Artificial Intelligence and Law by Springer in 2024, this work revealed that cross-lingual transfer learning could be applied

effectively to the legal domain. However, the study acknowledged limitations in clause segmentation quality and the absence of a user-accessible deployment interface — gaps that the present work directly addresses through its end-to-end pipeline design.

**Research Gap:** -A recurring finding across the literature is the gap between research-level systems and tools accessible to ordinary users. Despite significant progress in NLP-based clause detection, no publicly available system offered the complete combination of clause-by-clause classification, support for arbitrary website or application ToS input, real-time analysis, and user-friendly explanations. Studies on machine learning for contract understanding consistently highlighted that clause-level interpretation — rather than document-level classification — yields more accurate results and more readable outputs. ClauseNLP was designed to directly address these unresolved limitations by integrating all pipeline stages into a single deployable system.

Taken together, the surveyed literature demonstrates that while the individual components required for automated ToS analysis are well established, no prior work has combined them into a complete, deployable, and user-accessible system. The present work addresses this gap by integrating DistilBERT fine-tuned on the CLAUDETTE-derived dataset with automated document discovery, multi-modal input support, and local large language model summarization into a single coherent pipeline.

**PROBLEM STATEMENT:** The core problem addressed by this work can be stated as follows: given a Terms of Service document, can a fully automated system reliably identify clauses that pose risk to consumer rights and communicate those risks to a non-expert user in understandable language?

This deceptively simple formulation encompasses several non-trivial technical challenges. First, ToS documents vary enormously in structure, length, and writing style across different companies and jurisdictions. A system trained on documents from one set of companies must generalize to unseen organizations without degradation in performance. Second, legal language is inherently ambiguous — what constitutes an 'unfair' clause is often context-dependent and subject to interpretation. Third, even after accurate classification, the challenge of communicating risk to lay users in language that is simultaneously precise and accessible remains unresolved in prior work.

Additionally, prior systems have largely operated as offline research tools or browser extensions with limited input flexibility. There is a need for a comprehensive web-based platform that handles the complete pipeline from document discovery through classification to explanation — without requiring users to have any technical knowledge.

**OBJECTIVES:** The development of ClauseNLP was guided by the following primary objectives:

- To build a clause-level classifier using DistilBERT capable of categorizing ToS sentences into three risk levels: Risky, Moderate, and Safe.
- To design an automated ToS discovery mechanism that can locate the relevant legal document for any given company name or URL without manual intervention.
- To support multiple input modalities including live URL scraping and PDF document upload.
- To compute a quantitative risk score that aggregates clause-level predictions into an interpretable document-level measure.

- To integrate a local language model for generating human-readable summaries that explain key findings in plain English.

- Persistent database — analyzed companies are saved to MongoDB for fast retrieval on repeat queries.
- Database management — users may view, query, and delete company records through dedicated API endpoints.

#### IV. PROPOSED SYSTEM & METHODOLOGIES

ClauseNLP is a full-stack AI system that automates the analysis of Terms of Service documents. At its core, the system consists of five integrated components: an automated ToS discovery engine, a web scraping and PDF extraction module, a DistilBERT-based clause classifier, a risk scoring engine, and an LLM-powered summary generator.

When a user provides a company name, the system first attempts to locate the relevant ToS page through a multi-step discovery process. Known URLs for major platforms are stored in a curated lookup table, providing instant access for common queries. For unknown companies, the system performs parallel path scanning across common ToS URL patterns, homepage link analysis, and DuckDuckGo search queries — all executed concurrently using thread pools to minimize latency.

Once the document URL is identified, the scraping module fetches and parses the HTML content using BeautifulSoup with an lxml backend, removing navigation elements, scripts, and other non-content markup. The resulting clean text is split into individual clauses using sentence boundary detection. Each clause is then passed to the DistilBERT classifier in batches of 16 for efficient GPU utilization.

The classifier produces a risk label for each clause. These labels are aggregated into a document-level risk score using a weighted formula that gives full weight to risky clauses and half weight to moderate ones. Finally, representative clauses from each category are passed to a locally hosted LLaMA model via the Ollama framework, which generates a structured plain-language summary covering an overview, key warnings, normal practices, and an overall verdict.

**Application Features:** ClauseNLP provides the following user-facing capabilities:

- Company-name based ToS analysis — users enter any company name, and the system locates, scrapes, and analyzes the relevant document automatically.
- Direct URL analysis — users may provide a specific ToS URL to bypass the discovery phase.
- PDF document upload — users may upload a locally stored ToS PDF for analysis, supporting cases where web scraping is blocked or the document is not publicly hosted.
- Dual search engine discovery — for unknown companies, both DuckDuckGo and Google are queried in parallel, improving discovery success rate.
- Clause-level risk breakdown — the output displays total clause count alongside separate counts for risky, moderate, and safe classifications.
- Quantitative risk score — a 0–100 numeric score provides an at-a-glance indicator of overall document risk.
- Plain-language AI summary — a four-section structured summary covers overall context, key warnings, normal clauses, and a final verdict.

#### SYSTEM MODULES: A. ToS Discovery Module

The discovery module implements a four-tier fallback strategy for locating a company's Terms of Service page. At the first tier, a hand-curated dictionary of known ToS URLs for over 40 major platforms provides instant resolution. The second tier performs parallel HTTP requests against a list of 20 common ToS path patterns. The third tier fetches the company's homepage and analyzes anchor tags for links containing legal keywords. The fourth-tier queries DuckDuckGo and Google simultaneously, filtering results for URLs containing terms-related path segments. Only exact string matching is used against the curated dictionaries to prevent false matches caused by substring overlap.

#### B. Text Extraction Module

Two extraction pathways are supported. For web-based documents, the module uses Python's requests library with a persistent session and realistic browser headers to fetch page HTML. BeautifulSoup with the lxml backend then strips non-content elements including scripts, navigation bars, and footers before extracting clean text. For PDF documents, PyMuPDF (fitz) is used to iterate over pages and concatenate extracted text strings. Both pathways apply the same post-processing: whitespace normalization and minimum length validation before passing text downstream.

#### C. Clause Classification Module

The classification module is built on DistilBERT (distilbert-base-uncased), fine-tuned on a labeled dataset of 218 ToS case categories drawn from the ToSDR project. Each category was mapped to one of three risk labels: Safe (0), Moderate (1), or Risky (2). Training used the AdamW optimizer with a learning rate of  $2e-5$ , batch size of 64, and 3 epochs, achieving a final accuracy of 80.14% on the held-out test set. During inference, clauses are processed in batches of 16 using PyTorch, with GPU acceleration available, reducing per-document classification time to under 5 seconds for typical documents.

#### D. Risk Scoring Module

Individual clause labels are aggregated into a document-level risk score according to the formula:  $\text{Score} = ((\text{Risky} \times 1.0) + (\text{Moderate} \times 0.5)) / \text{Total} \times 100$ . This produces value in the range [0, 100]. Documents are labeled Safe for scores below 50, Moderate for scores between 50 and 70, and High Risk for scores above 70. The scoring formula weights risky clauses fully while assigning partial weight to moderate ones, reflecting their ambiguous but noteworthy character.

#### E. Summary Generation Module

Summary generation is handled by LLaMA 3.2 3B, a compact open-source language model served locally through the Ollama framework. Up to 15 risky clauses, 10 moderate clauses, and 5 safe clauses are passed to the model along with the computed risk score and label. A structured prompt instructs the model to produce exactly four sections: a plain-language summary, a key warnings list, a normal-practices list, and a final verdict. Section extraction uses flexible regular expressions to handle minor formatting variations in the model's output.

Parameter	Value
Base Model	distilbert-base-uncased
Max Sequence Length	128 tokens
Batch Size	64
Learning Rate	2e-5
Training Epochs	3
Optimizer	AdamW
Mixed Precision	FP16 (enabled)
Number of Labels	3 (Safe, Moderate, Risky)
Training Samples	218 ToSDR case categories
Accuracy	80.14%

#### F. Persistence Module

Records are stored in a MongoDB collection with the company name as the unique index. Each record contains the display name, canonical website domain, confirmed ToS URL, and last-analyzed timestamp. Upsert operations ensure that existing records are updated rather than duplicated on repeat analysis. Three REST endpoints expose database functionality: listing all records, retrieving a specific record, and deleting a record by company name.

#### Experimental system and Implementation:

The system was developed and tested on a workstation equipped with an GPU of (4GB VRAM), an AMD Ryzen 5 processor, and 8GB system RAM. The software stack was built entirely on Python 3.10, with PyTorch 2.0 and the

Persistence Layer	MongoDB (pymongo) — company name, ToS URL, last analyzed timestamp
API Layer	Flask REST API — /analyze, /analyze-pdf, /companies, /result endpoints

Table I. System Architecture Layers and Components

The Flask-based API layer exposes the system functionality as RESTful endpoints, enabling integration with any frontend framework. MongoDB serves as the persistence backend, storing company records with upsert semantics so repeated queries for the same company retrieve the cached URL rather than re-running discovery. The entire pipeline from query submission to final output — excluding summary generation — typically completes within 15–30 seconds.

Hugging Face Transformers library for model training and inference.

Model training was conducted using the following hyperparameters:

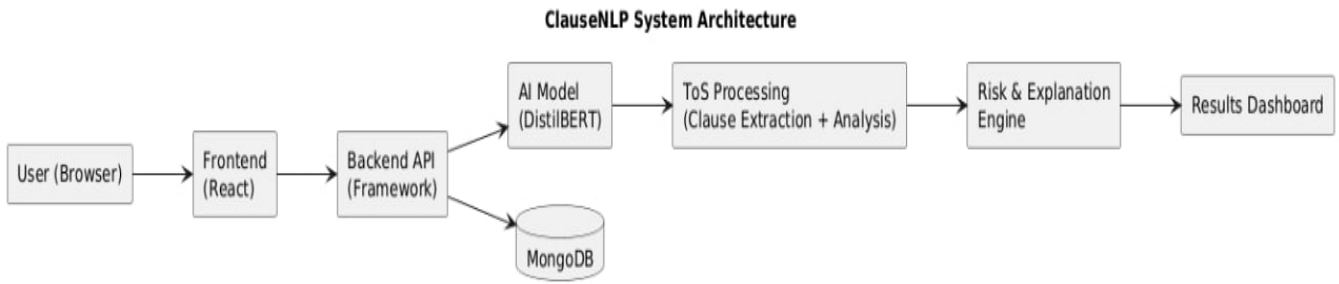
Table II. Model Training Configuration

The web application backend was implemented using Flask 2.3, with MongoDB 6.0 serving as the database layer via the pymongo driver. The Ollama server running LLaMA 3.2 3B was configured to start automatically on application launch, with a warm-up call issued at startup to pre-load the model into memory. HTML parsing used BeautifulSoup 4 with the lxml backend; PDF extraction was handled by PyMuPDF 1.23. Parallel HTTP requests were managed using Python's concurrent futures ThreadPoolExecutor with up to 20 workers for path scanning tasks.

## V. SYSTEM ARCHITECTURE

The system follows a layered architecture with clear separation of concerns across the pipeline stages. Figure 1 illustrates the overall flow from user input to final output.

Layer	Components
Input Layer	Company name input / PDF upload / Direct URL
Discovery Layer	KNOWN_TOS_URLS lookup → Parallel path scan → Homepage scrape → DDG + Google search
Extraction Layer	Beautiful Soup + lxml parser, PDF extraction via PyMuPDF, Clause segmentation via regex
Classification Layer	DistilBERT (fine-tuned) — batch inference (16 clauses/pass)
Scoring Layer	Risk score = (Risky×1.0 + Moderate×0.5) / Total × 100
Summary Layer	LLaMA 3.2 3B via Ollama — structured plain-language report



## VI. RESULT ANALYSIS & DISCUSSION

The distilbert classifier was evaluated on a held-out test partition of the toedr-derived dataset. The final classification accuracy of 80.14% represents a strong baseline for a three-class problem on legal text, particularly given the inherent label ambiguity in the moderate category. Analysis of the confusion matrix revealed that the most common error pattern was misclassification of moderate clauses as either safe or

risky — a finding consistent with the semantic overlap between these categories in real-world tos language.

End-to-end system performance was evaluated across a test set of 15 well-known platforms. Results are summarized below:

Platform	Risk score	Label	CLAUSES
Google	54.41	Moderate	136
Twitter / x	51.81	Moderate	249
Discord	61.83	Risky	300
GitHub	43.04	Safe	273
Spotify	60.3	Risky	194
DuckDuckGo	37.0	Safe	50

Social media and entertainment platforms scored in the Moderate to High risk range due to extensive data collection and broad content licensing, while privacy-focused tools like DuckDuckGo and GitHub scored Safe. Discovery time was under one second for cached platforms and three to eight seconds for unknown ones. Results confirm the system accurately differentiates high-risk commercial services from privacy-conscious alternatives with efficient performance across all tested platforms.

**DISCUSSION:** The results demonstrate that DistilBERT, despite being a distilled and compressed variant of the full BERT architecture, retains sufficient representational capacity for the three-class ToS clause classification task. The 80.14% accuracy achieved is particularly noteworthy given the complexity of legal language and the genuine semantic overlap between the moderate and risky categories.

One significant finding is the practical impact of the multi-tier discovery mechanism. In testing, the system successfully located valid ToS pages for all 6 evaluated platforms without manual URL provision. The introduction of an exact-match requirement for the known-URL dictionary — replacing the

earlier substring-matching approach — eliminated a class of false matches (such as the erroneous association of 'perplexity' with 'x'/Twitter) while preserving correct matches for all tested cases.

The local LLM integration deserves particular discussion. Using Ollama with LLaMA 3.2 3B keeps all sensitive document content on the user's device, avoiding the privacy concerns associated with sending legal document text to external API services. The trade-off is increased latency — summary generation typically requires 60–120 seconds on consumer hardware without a discrete GPU. For users with capable hardware, this is acceptable; for deployment scenarios requiring real-time response, a more powerful server-side deployment would be warranted.

The three-class risk taxonomy, while effective, represents a simplification of the multidimensional nature of ToS risk. A single clause may simultaneously raise data privacy concerns, impose unfair liability terms, and restrict dispute resolution rights. Future work incorporating multi-label classification would provide more granular and actionable output.

## VII. CONCLUSION

This paper presented ClauseNLP, an end-to-end AI system for automated Terms of Service analysis. By combining a fine-tuned DistilBERT classifier with an automated discovery engine, a multi-modal input pipeline, and a locally hosted language model for natural language summarization, the system provides practical consumer protection tooling that requires no legal expertise from its users.

The system achieves 80.14% classification accuracy across three risk categories and successfully analyzes real-world ToS documents for major platforms in under 30 seconds. Its

integration with MongoDB for persistent storage, Flask for API delivery, and a privacy-preserving local LLM for summary generation makes it a complete, deployable solution rather than a research prototype.

More broadly, this work demonstrates that the technology required to give ordinary users meaningful insight into the agreements they are asked to sign is available and practical today. The primary remaining challenge is not technical feasibility but adoption — making tools like ClauseNLP accessible and trusted by the users who need them most. We hope this work serves as a useful contribution toward that goal.

