BANDHOO: A Safety-Constrained Decision-Support System for Bilingual Police Complaint Drafting, Statutory Mapping, and Jurisdiction Assignment under the Bharatiya Nyaya Sanhita

Tanaya Paralkar
Department of Data Science
Mukesh Patel School of
Technology Management and
Engineering, NMIMS
Mumbai, Maharashtra, India
tanayaparalkar@gmail.com

Nitaanth Panickar
Department of Data Science
Mukesh Patel School of
Technology Management and
Engineering, NMIMS
Mumbai, Maharashtra, India
nitaanth.panickar@gmail.com

Viraj Darekar
Department of Data Science
Mukesh Patel School of
Technology Management and
Engineering, NMIMS
Mumbai, Maharashtra, India
darekarviraj2@gmail.com

Jaya Jeswani
Department of Data Science
Mukesh Patel School of
Technology Management and
Engineering, NMIMS
Mumbai, Maharashtra, India
jaya.jeswani@nmims.edu

Giridhar Urkude
Symbiosis Institute of Technology,
Nagpur Campus,
Symbiosis International
(Deemed University)
Pune, India
giridharurkude@gmail.com

Abstract **- Citizens of India often struggle to articulate their crime report in terms of relevant legal jargon, and find it difficult to ascertain the correct police station in relation to the crime, and whether it is cognizable. We present BANDHOO, a decision support prototype that can generate a structured, multilingual draft complaint and a pre-addressed email complaint from an open-text complaint. BANDHOO is not an autonomous legal decision-maker and never submits or registers any First Information Report (FIR). We introduce a decoupled scoring function that results in (i) a contradiction gate returning a zero score when the statutory provisions have been negated, therefore achieving a result of 0.00% positive false charges, (ii) a multi-action max-pooling function operating at the clause level to address dominant offence masking, thereby achieving a candidate recall rate of 65.62%, (iii) a precise span grounded extraction methodology resulting in 2.56% ungrounded facts on heldout cases, (iv) a point-in-polygon jurisdiction identification, with a buffer of 100m, achieving zero unsafe assignments across our spatial benchmarks and (v) a transparent deterministic fall-back function that ensures the service remains available when upstream models fail with latency of approximately 12ms on production. We also present a taxonomy of error types which we categorise into five types. We present the limitations of the study including synthetically written ground truth data and eight benchmark cases in a preliminary status as they require review by advocates.**

Key words **- Legal Information Retrieval, Retrieval Augmented Generation, Bharatiya Nyaya Sanhita, Grounded Extraction, Geographical Jurisdiction, Human in the Loop, Public Sector AI Safety.**

I. INTRODUCTION

Three non-trivial decisions must be made by a lay person to submit a report; which statutory provision(s) apply to the situation, whether it is an offence, and which station has territorial jurisdiction. This can result in delays, complaints being submitted incorrectly and/or under-describing. However, the Supreme Court of India has decided that if any information relating to a cognisable offence is disclosed to the police, they have an obligation to record an FIR [1]. The actual offences are now contained in the Bharatiya Nyaya Sanhita 2023 (BNS) [2]. Thus, this is a genuine justice accessibility issue.

Large Language Models (LLM) can assist but suffer from hallucinations, providing incorrect and untrue facts, and confidently providing incorrect jurisdiction. This can cause confusion for the complainant [3]. BANDHOO addresses this by integrating the LLM into a pipelined system, making any safety-critical decisions via deterministic rules. The generated draft will not be submitted or confirmed.

Contributions

1.Negative provisions are not scored due to decoupled scoring (Section A).
Multi-offence complaints:
2.Max pooling with multi-actions at clause level.
3.Extracted facts that are exact (character offset verified).
4.Buffering for geographical jurisdiction with abstention on proximity to station.
5.Offline deterministic method for critical fallback.

No novel architecture or foundation model is presented. However, this work provides valuable insights into safe

system design, ensuring a workflow in a critical area of public administration, for public welfare. An honest error analysis is carried out.

II. Related Work

Legal Information Retrieval

Statutes and case law have been heavily researched in the COLIEE competition series [4]. Lexical models like Okapi BM25 [5,6] provide a strong baseline for this task. While Retrieval Augmented Generation (RAG) [7] ensures generation is grounded in the passed in retrieved information, this does not allow discrimination between positive and negative provisions, motivating our decoupled scoring.

Legal NLP

Domain trained encoders such as LEGALBERT [8] and surveys on AIin law [9] show it's possible to understand legal texts. Within Indian law, the ILDC corpus can be used for judgement prediction and explanation [10]. These resources are aimed at judges and researchers, rather than front-line complainants and do not fully address character span level attribution constraints.

Safe AI in Public Administration

Due to concerns around unintended behaviours in deployed systems [11] and hallucinations [3], Bandhoo is designed to defer to humans. Any situations with insufficient evidence (statutory provision requires information not provided by the complainant) and ambiguous geographical locations will be deferred.

Geographical Analysis

Ray casting [12,13] is a classical approach to point in polygon analysis. In this work we use hole exclusion and buffering by metric to ensure any uncertainty in boundaries results in abstention.

Table 1. Literature Streams and the Gap Addressed

Stream	Representative work	Gap addressed by Bandhoo
Legal IR	COLIEE [4], BM25 [5, 6]	Ranking is insensitive to negation of a provision in the narrative
Legal NLP	LEGAL-BERT [8], ILDC [10]	No character-level attribution of extracted facts; audience is courts, not complainants
Safe AI in public administration	Registration duty [1], deferral [11]	Procedural abstention for provisos and ambiguous territory
GIS analysis	Ray casting [12,13]	Boundary uncertainty should yield abstention, not a nearest-centroid guess

Table 1 organises these streams and states the gap that each leaves open. We position Bandhoo as a decision-support prototype built from established components; its novelty is in how those components are constrained and composed, not in any single model.

III. Problem Formulation and Design Principles

Given a free form narrative x (either in English or the complainant's language), location ℓ (optional) and collection of evidence files E, Bandhoo produces a tuple, (P,F,z,D,M). P is a ranked list of BNS provisions, with valid elements. F is a collection of structured facts, each with a source span. z is the police station, or null. D is a bilingual draft complaint and M is an e-mail draft. Everything is suggested.

As this is a high stakes domain, we ensure a number of invariants are held throughout the implementation (Table 2). Each invariant is enforced using deterministic code/testing.

Table 2. Safety Invariants and How They Are Enforced

ID	Invariant	Enforcement
I1	Output is a draft; never submitted, confirmed, or an FIR	Client notices; trust-safety test suite
I2	A contradicted provision cannot rank: $\chi = 1 \Rightarrow S = 0$	Decoupled scoring, Eq. (1)
I3	Every fact is a verbatim span of the narrative	Span check, Eq. (3); rejection of unsupported facts
I4	Ambiguous territory yields no station	100m buffer; station: null
I5	Unverified backend output cannot override local containment	Client-side precedence rule; test
I6	Backend failure shows no legal sections	Empty-state rendering; test
I7	Availability under upstream outage	Deterministic offline fallback

IV. System Architecture

Fig. 1 shows the implemented five-layer design: a client tier; an API tier; a legal engine (retrieval, scoring, jurisdiction, evidence screening); a LangGraph [14] orchestration layer for structuring; and a PostgreSQL persistence layer accessed through SQLAlchemy 2.0.

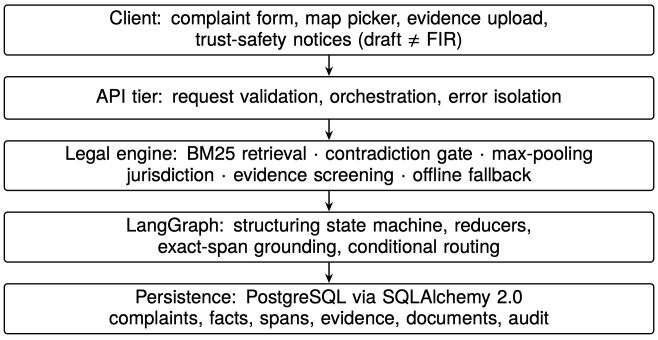


Figure 1. Implemented system architecture.

ORM models are used, these are; ComplaintModel, MissingPersonModel,StructuredFactModel,FactSourceSpanModel,EvidenceFileModel,VerificationResultModel, GeneratedDocumentModel,DocumentVersionModel, EmailDraftModel, AuditEventModel. The initial migration is used to create the indexes, foreign keys (enforce cascading deletions) and structured payloads using JSON data type. The pipeline decisions are stored and an audit event table is utilised.

Outputs are a bilingual ReportLab PDF with a legal disclaimer and station address. An e-mail draft is created following the format ps.mum@mahapolice.gov.in. These are not auto-generated.

A. End-to-End Complaint Pipeline

The screaming track shows the process from intake to output (Fig 2). The intake includes the story, location and evidence. The legal engine predicts the provisions and

parallelly filters and resolves evidence and jurisdiction. The LangGraph provides a structure to the facts and the output generates the PDF and e-mail drafts. The decisions throughout are logged in the audit table.

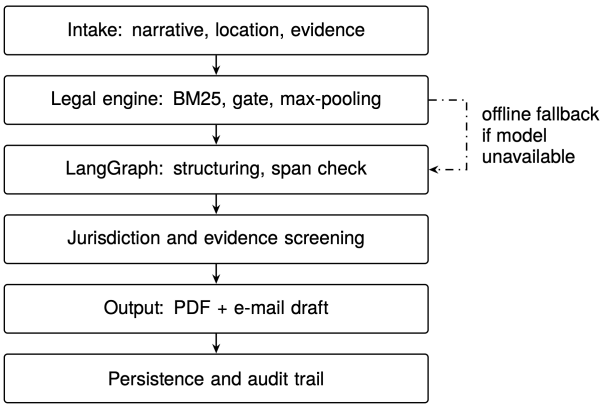


Figure 2. End-to-end complaint pipeline from intake to persisted draft outputs.

B. Document Generation & Dispatch

A short description of the pipeline. The API requests the PDF from the complaint engine. The ReportLab creates a bilingual draft with a legal disclaimer and station address. This is saved along with a new version of the document. An e-mail is created by the e-mail module. The user will manually check and dispatch. All decisions made throughout the pipeline are logged into the audit event in persistence using the ORM models listed above.

V. Methods

A. Retrieval & Decoupled Scoring

Okapi BM25 [5] is used to index and rank the statutory provisions. Given a narrative x and provision p, lexical scoring is performed with score BM25(x,p). A negation lexicon and patterns N are used to calculate contradiction indicators $\chi_p(x,N) \in \{0,1\}$, given the narrative x. The final score is calculated as;

$$S_{\text{final}}(x,p) = (1 - \chi_p(x,\mathcal{N}))\, S_{\text{rel}}(x,p) \quad (1)$$

Where $\chi_p = 1$ results in $S_{\text{final}} = 0$. This decouples relevance and contradiction, preventing the lexical score from outweighing an explicit contradiction. All elements of each offence in the candidates are validated.

B. Clause Level Multi Action Max Pooling

Complaints often mention multiple actions (e.g. trespass and theft). Scoring the whole narrative could cause the main offence to dominate. This is overcome by splitting the narrative into action clauses $a_1 \ldots a_K$ and using max pooling;

$$S_{\text{pool}}(x,p) = \max_{1 \le k \le K} S_{\text{final}}(a_k,p) \quad (2)$$

This results in a provision being identified if there is a clause supporting it. Equation (1) removes any contradicting clauses.

The full ranking algorithm is shown in Algorithm 1. The contradiction indicators are used in the clause loop and a proviso cannot be added on in a later step if the indicator is negated.

Algorithm 1 Decoupled, clause-pooled provision ranking

Require: narrative x, provision index P, negation patterns N, list size n
Ensure: ranked candidates C with element status
$A \leftarrow$ SegmentClauses(x)
for all $p \in P$ **do**
  $S[p] \leftarrow 0$
  for all $a_k \in A$ **do**
    $r \leftarrow$ BM25(a_k, p)
    $\chi \leftarrow$ Contradicts(a_k, p, N)
    $s \leftarrow (1 - \chi) \cdot r$ {Eq. (1)}
    $S[p] \leftarrow \max(S[p], s)$ {Eq. (2)}
  end for
end for
$C \leftarrow$ TopN(S,n) keeping only $S[p] > 0$
for all $p \in C$ **do**
  if not ElementsSatisfied(x, p) **then**
    mark p as needing more facts; request missing items
  end if
end for
return C

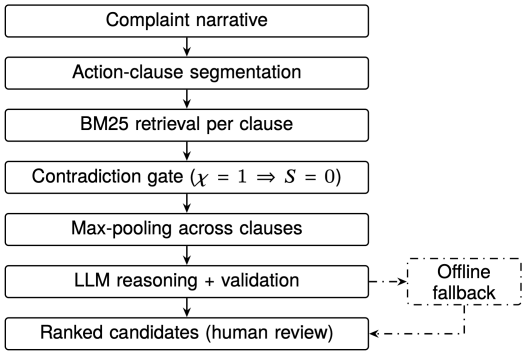


Figure 3. Legal retrieval flow. The dashed branch is taken when the upstream model is unavailable.

C. Structured Using LangGraph & Exact Span Grounding

The structured is implemented using a StateGraph from LangGraph [14]. Reducers, such as keep_original, protect the state fields from being overwritten by subsequent nodes. Conditional edges are used to skip around failed nodes. Error isolation guards ensure that any failed nodes do not cause the run to become corrupt. For each fact f obtained, a source span is attached (i,j), and they are only accepted if:

$$T[i{:}j] \equiv \text{text}(f) \quad (3)$$

Where T is the original narrative. If they fail to meet the requirement in equation (3), they are rejected as they cannot be supported. These spans are saved to the FactSourceSpanModel so the each fact can be verified to come from the complainant's own words.

Why reducers are important. By having a naive state in the graph which is based on last-write-wins, we allow an override of the values. The keep_original reducer ensures that the state remains monotonic. This means nodes can be added to the verified state but not overwritten. This, along with the conditional edges and error guards, ensures that the failure of any node only affects that node and there is no potential for partial results to appear as complete.

The grounding rule and monotonic merge of the reducer is shown below.

Algorithm 2: Grounding Rule.

Algorithm 2 Exact-span fact acceptance and monotone merge

Require: narrative T, candidate facts $F = \{(key,text,i,j)\}$, state Σ
Ensure: updated state Σ, rejected facts R
$R \leftarrow \varnothing$
for all $(key,text,i,j) \in F$ **do**
if $0 \le i < j \le |T|$ and $T[i:j] = text$ **then**
if $\Sigma[key]$ is empty **then**
$\Sigma[key] \leftarrow (text,i,j)$ {keep_original}
end if
persist span (i,j) for the fact
else
$R \leftarrow R \cup \{key\}$ {unsupported}
end if
end for
return Σ, R

D. Jurisdiction Buffering

We first perform a naive ray casting (Jordan Curve Theorem) [12,13] to determine whether a point q is contained in a region with no holes. We then remove holes. We project the coordinates to measure how far away from the boundary using an equirectangular approximation of the local reference latitude (ϕ_0):

$$x = R(\lambda-\lambda_0)\cos(\phi_0),\ y = R(\phi-\phi_0) \quad (4)$$

Where R is the mean radius of the Earth. If the point q is within 100m of the boundary, we do not assert any stations. Instead,station: null is returned, and we ask the user if this is correct. This is similar to the benefit of a Zero FIR transfer. We do not want to provide any incorrect confident assignments. We cannot override the local containment on the client side with an unverified backend prediction.

This provides a strict geometric decision, and can return three outcomes: inside, outside or ambiguous. This means that just because a point is close to a municipal line, it does not imply that it would be placed on one side without anyone knowing. The cost of performing no action (one extra confirmation from the user) is much lower than confidently providing incorrect assignments (which can be a Zero FIR).

The assignment rule is shown below. This includes the ray casting predicate and buffering:
Algorithm 3: Assignment Rule.

Algorithm 3 Buffered jurisdiction assignment

Require: location q, stations $\{(id,outer,holes)\}$, buffer $\delta = 100$m
Ensure: station id or null
project all coordinates to metres around $\phi_0 = \text{lat}(q)$
$inside \leftarrow \varnothing$; $d_{\min} \leftarrow \infty$
for all station s **do**
if RayCast$(q,s.outer)$ and not $\exists\ h \in s.holes$: RayCast(q,h) **then**
$inside \leftarrow inside \cup \{s\}$
end if
$d_{\min} \leftarrow \min(d_{\min}, \text{DistToBoundary}(q,s))$
end for
if $|inside| \neq 1$ or $d_{\min} \le \delta$ **then**
return null {ask user to confirm}
end if
return the single station in *inside*

The RayCast predicate is the most common even_odd (Jordan curve) predicate. It is defined in [12,13].

E. Evidence Filtering

The uploaded files are filtered using the following rules: (i) magicbyte type checking, (ii) byte variance sharpness heuristic, (iii) generative tool meta detection using a Latin-1 regular expression, and (iv) 64-bit difference hash (dHash) [15] to find near duplicates. These are filtering heuristics and never use the word "forensic".

The dHash is calculated by scaling down the grayscale image and calculating the neighbouring pixel intensities. This information is stored in a 64-bit hash. Two files with a small Hamming distance between their hash values are deemed potential duplicates. The outcome of the verification is stored in the VerificationResultModel on a per file basis. This can be stated in the interface without implying they were authenticated. The order of filtering is shown below:
Algorithm 4: Filtering Order.

Algorithm 4 Evidence screening

Require: uploaded file bytes b, stored signatures H, Hamming threshold τ
Ensure: screening record (not forensic proof)
if not MagicBytesValid(b) **then**
return record with type check failed
end if
$sharp \leftarrow$ ByteVarianceSharpness(b)
$gen \leftarrow$ Latin1Scan(b) for generative-tool markers
$h \leftarrow$ dHash64(b)
$dup \leftarrow \exists\ h' \in H$: Hamming$(h,h') \le \tau$
$H \leftarrow H \cup \{h\}$
return record$(sharp,gen,dup)$

F. Deterministic Offline Pathway

If there is no upstream LLM, or the quota is reached, we use a rule based pathway. This uses the same BM25 index, contradiction gate,and span based extraction to produce a draft without any external dependencies. If the fallback option is enabled, this is indicated to the user. The median latency is ~12ms.

This is a summary of the steps and where the fall-back is used:

Algorithm 5 Complaint processing with offline fallback

Require: complaint (x,ℓ,E)
Ensure: draft D, e-mail draft M, audit trail
$z \leftarrow$ AssignStation(ℓ) {Alg. 3}
screen each $e \in E$ {Alg. 4}
try $P \leftarrow$ LiveRank(x)
except model unavailable or quota exhausted:
$P \leftarrow$ OfflineRank(x) (Alg. 1 with offline rules); flag fallback mode
$F \leftarrow$ Structure(x) with span check {Alg. 2}
$D \leftarrow$ CompilePDF(P,F,z) with draft-only disclaimer
$M \leftarrow$ PrepareEmail(D,z)
write audit events for every decision
return D,M

VI. Evaluation Methodology

We split the benchmark cases into development and holdout sets. We run them in both Live and Fallback modes. We report

the false positive rate for charges, candidate recall for production, unsupported fact rate for structuring, unsafe assignment rate for jurisdiction, and the fallback latency. We run the front-end trust-safety suite (23 tests) to verify that notices are reported that the draft wasn't submitted, confirmed or an FIR. Any missing backend will be represented by not having any fabricated sections. Any near border points will require a confirmation.

Ground Truth

The project team have created a synthetic gold set of annotations. There are still 8 cases to finalise.

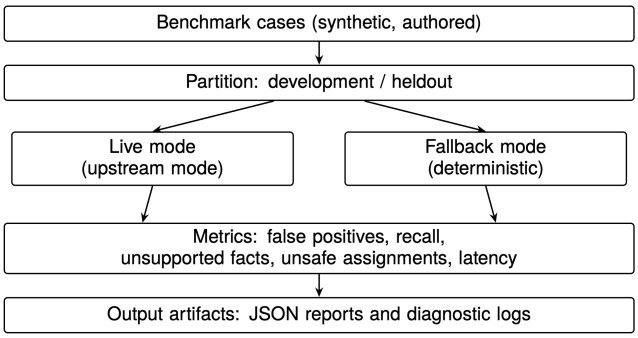


Figure 4. Evaluation methodology: dataset partitioning, dual inference modes, metrics, and output artifacts.

A. Trust-Safety Verification Frontend

The client was successfully authenticated using a specific test suite (node src/trustSafety.test.js). All 23/23 tests passed in 95ms. A summary of the behaviour is shown below:

VII. Results

Table 3. Summary of Measured Results

Metric	Value	Component
False positives on contradicted charges	0.00%	Decoupled scoring
Candidate recall (production)	65.62%	Max-pooled retrieval
Unsupported facts (heldout)	2.56%	Span grounding
Unsafe territorial assignments	0.0%	Buffered containment
Fallback latency	≈12ms	Offline fallback
Frontend trust-safety tests	23/23	Client safeguards

Contradicted charges have a false positive rate of 0%. This is due to the construction of Eq. (1). Whether or not we have detected a contradiction, there is no risk of getting a ranked charge. We do not know what the detection recall will be (Section VIII). Having a candidate recall of 65.62% makes it clear that the retrieval is not the final classifier for finding candidates. An abstention policy was applied. 0 unsafe assignments means that there are intentionally no points that aren't being assigned.

B. Component Interpretation

Decoupled Scoring

A contradiction was detected so it is not a ranked charge. We are guaranteed there is a contradiction. There is no risk of ranking errors. Therefore category 2 in table 4 is more important.

Max Pooling

The more clauses are pooled, the higher the chances of secondary offences being surfaced. However, we have less candidates for the reader to consider. The element validation step checks for one noisy clause. This is because Eq. (2) uses a max.

Span based grounding

There is an unsupported fact rate of 2.56% for the holdout cases. This means nearly all facts extracted from the holdout cases have a comparison to the complainant's description. All facts that have been accepted have a corresponding span which the reviewer can view.

Jurisdiction

The zero unsafe assignment rate isn't an accuracy rate. The percentage of points covered should be stated in the final tables (points covered - percentage of points covered by a station).

Fallback

There is a fallback pathway available even if the network is down (~12ms). The results are labelled as fallback and are assessed separately.

Table 4. Error Taxonomy and Mitigations

No.	Category	Trigger	Mitigation
1	Statutory provisos	Property value unknown (BNS §303(2) proviso) blocks cognizability call	Ask for value; defer
2	Grievance-gate leakage	Civil/criminal mix, e.g., BNS-BM-48 contractor overcharging with “cheated”	Stricter non-penal gate; review
3	Syntactic inversion	Agentless passive (“padlock was found smashed”) missed by active-verb extractors	Passive patterns in extractor
4	Multi-offender thresholds	Dacoity (BNS §310) needs ≥ 5 actors; BM17 has 2	Count actors; downrank
5	Territorial edge ambiguity	Within 100m of boundary	Abstain; user confirms

Table 4 gives the five-category taxonomy derived from the diagnostic logs of the fallback evaluation and the structuring benchmark.

VIII. Error Analysis

Category 1: Correct action is to look for missing facts. This means there is a legitimate uncertainty if there is a statutory proviso. This isn't an error in the model. Category 2: False negative observed by non-penal gate (BNS-BM-48). Category 3: Missed action items in holdout structuring cases (H04, H13). Category 4: Precision risk. There is lexical overlap with 'dacoity' vocabulary which could encourage a category of provision. Category 5: Willful and safe abstention.

A. Case Studies

Proviso (Cat. 1): The proviso in BNS §303(2) can be used to state that theft is cognisable based on the value of the property. The system may list the missing property and defer. *Gate Leakage (Cat. 2):* Over the counter charge by a contractor. The story says 'cheated'. Although this is a commercial complaint, the crime vocabulary leaks through the non-penal gate. *Passive voice (Cat. 3):* Padlock found

smashed. This isn't picked up by any active verb extractors. (Holdout cases H04 & H13). *Threshold (Cat. 4):* Some lexical overlap but no numeric element in dacoity (BNS §310). *Edge (Cat. 5):* System returns station: null (intentionally no answer) within 100m of a boundary.

IX. Discussion

These results validate that if you ask a legal drafting assistant to enforce explicit invariants, it will not make any assertions it cannot ground. They do not mention anything about the legality of the field due to the ground truth being synthetic. They all have safeguards in place to ensure coverage (contradicted charges are dropped, unsupported facts are denied, ambiguous territory has no station, provisos result in a deferral). We believe this is the correct default rule for coverage if the output is a contact with the criminal justice system by a citizen. The design considers the location of where information about cognisable offence should lead to registration [1],and Bandhoo will never attempt to discourage FIR registration.

X. Limitations and Threats to Validity

- *Ground Truth:* The rates do not have external validity as the authors have indicated this in the annotations. There are 8 more cases to be reviewed by advocates. Point estimates are unreliable in small samples. Confidence intervals are required.
- *Method:* Paraphrasing (65.62% candidate recall) and unusual phrasings (Category 2 & 3) are not captured by BM25. The 0.0% unsafe assignment rate is partially due to the abstention. Coverage should be reported.
- *Scope:* Only covers modelled BNS provisions and station boundaries in Mumbai. Evidence filtering is heuristic.
- *Deployment:* No integration with any official registries. No user studies with complainants.

XI. Ethical Considerations

Not an FIR, Not Legal Advice. All outputs are labelled as 'draft' meaning they have not been submitted or verified. The system does not call the police. *Oversight.* Wonks in provisos, unclear geography and low grounded facts. No section of backbone is not available. *Privacy.* They will need to ensure that the complaints contain sensitive data, only store them for drafting, delete them in the event of failure at any stage and have a clear audit trail. If deployed they will need to be encrypted, have access rights and only store them for a limited amount of time. They will also need to comply with the Indian data protection law. *Fairness.* If the system is optimised for synthetic data, it may not serve dialects or code mixed text as well. This needs to be quantified before deployment. *Misuse.* A fluent draft may help with false complaints. There is no verification of facts in the system. Disclaimers and non-forensic evidence labels stop over-trusting.

XII. Future Work

Future work will look at using dense (bi-encoder) retrieval alongsideBM25. Adapters for official case management interfaces, which require authorisation. Learned image forgery detection alongside the current heuristics. Gold set of cases which have been advocated by legal experts. A usability study with complainants. Expansion to other cities and statutes.

XIII. Conclusion

Bandhoo shows that a legal complaint assistant can be created using explicit safety invariants. Contradicted charges have a zero score, facts are included as quotes from the narrative, ambiguous geography leads to abstention, and the system becomes a clear offline pathway. This provides a balance between recall and trustworthiness. This is acceptable as the output will be a citizen interacting with the criminal justice system. The numbers are preliminary and based on synthetic and partially provisional ground truth. They should be accessed independently and in the field.

References

[1]*Lalita Kumari v. Government of Uttar Pradesh*, (2014) 2 SCC 1, Supreme Court of India.

[2]Government of India, *The Bharatiya Nyaya Sanhita, 2023*, Act No. 45 of 2023.

[3]Z. Ji *et al.*, "Survey of hallucination in natural language generation," *ACM Comput. Surv.*, vol. 55, no. 12, art. 248, 2023.

[4]J. Rabelo, M.-Y. Kim, R. Goebel, M. Yoshioka, Y. Kano, and K. Satoh, "Overview and discussion of the competition on legal information extraction/entailment (COLIEE) 2021," *Rev. Socionetwork Strategies*, vol. 16, pp. 111–133, 2022.

[5]S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford, "Okapi at TREC-3," in *Proc. 3rd Text REtrieval Conf. (TREC3)*, NIST SP 500-225, 1995, pp. 109–126.

[6]S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, pp. 333–389, 2009.

[7]P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 9459–9474.

[8]I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The muppets straight out of law school," in *Findings of EMNLP*, 2020, pp. 2898–2904.

[9]H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu, and M. Sun, "How does NLP benefit the legal system: A summary of legal artificial intelligence," in *Proc. ACL*, 2020, pp. 5218–5230.

[10]V. Malik *et al.*, "ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation," in *Proc. ACL-IJCNLP*, 2021, pp. 4046–4062.

[11]D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," arXiv:1606.06565, 2016.

[12]M. Shimrat, "Algorithm 112: Position of point relative to polygon," *Commun. ACM*, vol. 5, no. 8, p. 434, 1962.

[13]E. Haines, "Point in polygon strategies," in *Graphics Gems IV*, P. S. Heckbert, Ed. San Diego, CA, USA: Academic Press, 1994, pp. 24–46.

[14]LangChain Inc., "LangGraph," software documentation. [Online]. Available: https://github.com/langchain-ai/langgraph

[15]N. Krawetz, "Kind of like that," Hacker Factor Blog, Jan. 2013. [Online]. Available: https://www.hackerfactor.com/blog/