Traceable by Design: An LLM Pipeline and Dashboard for EU Regulatory Consultation Analysis*

Thales Bertaglia[1,*], Haoyang Gui[1], Catalina Goanta[1] and Gerasimos Spanakis[2]

[1]Utrecht University, The Netherlands
[2]Maastricht University, The Netherlands

Abstract

Public consultations generate large volumes of data in the form of stakeholder submissions that are practically unfeasible to analyse manually. We present an end-to-end LLM-based pipeline and interactive dashboard for structured topic extraction from regulatory consultation submissions, demonstrated on the European Commission's Digital Fairness Act (DFA) public call for evidence as a case study. The system processes raw PDF attachments and web-form responses, extracts topic annotations, and grounds every extraction in a verbatim quote from the source text. Applied to 4,322 DFA submissions, the pipeline produced 15,368 topic annotations supported by 20,951 verbatim evidence quotes. Three principles govern the proposed design: verbatim grounding, full traceability, and transparency by design. The dashboard exposes the full extraction dataset through five analytical views, from dataset-level topic overviews to individual paragraph drill-downs, with every result traceable to its source. Beyond the predefined DFA topic categories, the pipeline generated certain stakeholder concerns, such as Age Verification, Payment Processor Censorship, and Digital Ownership, that a fixed-taxonomy approach would have missed. The pipeline is domain-generic; adapting it to a new consultation requires only a prompt update and a new dataset. A live demo is available at https://dfa-dashboard.thalesbertaglia.com/. The code and processed data are publicly available at https://github.com/thalesbertaglia/dfa-dashboard.

Keywords

public consultation analysis, evidence extraction, large language models

1. Introduction

EU public consultations and calls for evidence represent a formal mechanism of evidence-based policymaking. Before major regulatory reforms are adopted, the European Commission invites citizens, businesses, civil society organisations, and academic institutions to submit written responses via the "Have Your Say" portal [1]. These submissions constitute a form of structured public testimony: stakeholders articulating positions, raising concerns, and providing supporting arguments that regulators must take into account in the democratic decision-making process. As consultations generate large forms of data in volume and visibility, it becomes practically infeasible to analyse all submissions manually.

The public consultation on the EU Digital Fairness Act (DFA) illustrates the challenge. The DFA is expected to update the EU consumer acquis, and as such has attracted a lot of attention from the public at large. Between July and October 2025, the consultation attracted 4,325 responses covering topics such as dark patterns, addictive design, unfair personalisation, misleading influencer marketing, and exploitative subscription practices. The submissions are heterogeneous in form: 94.1% are short web-form responses submitted directly via the portal, while the remaining 5.9% include multi-page PDF attachments that elaborate on the feedback. Taken together, they constitute a large dataset that is challenging to systematically review at scale without computational support.

* *RELAF'26: Reasoning with Evidence in Law Enforcement and Forensics, Workshop at ICAIL 2026, Singapore.*
*Corresponding author.
t.f.costabertaglia@uu.nl (T. Bertaglia); h.gui@uu.nl (H. Gui); e.c.goanta@uu.nl (C. Goanta); jerry.spanakis@maastrichtuniversity.nl (G. Spanakis)
0000-0003-0897-4005 (T. Bertaglia); 0009-0000-6232-4156 (H. Gui); 0000-0002-1044-9800 (C. Goanta); 0000-0002-0799-0241 (G. Spanakis)

To enable systematic analysis at this scale, we present an end-to-end pipeline and interactive dashboard for structured evidence extraction from regulatory consultation submissions. The system transforms raw PDF and web-form documents into auditable evidence records: for each submission, it uses LLMs to extract the topics raised and the verbatim passages that support each topic claim. Applied to the preprocessed and filtered DFA dataset, the pipeline processed 4,322 documents, producing 15,368 topic annotations grounded in 20,951 verbatim evidence quotes. The DFA consultation serves as a case study; the pipeline is domain-generic and can be extended to other consultations and policy domains. Prior computational approaches to consultation analysis have focused primarily on document-level clustering and topic categorisation [2, 3]. These methods answer aggregate questions about what topics appear across the corpus, but do not produce structured, source-grounded evidence records: they cannot identify which specific passage in which specific submission supports a given position, nor provide a traceable link from an extracted claim back to its verbatim source. Filling this gap is the central contribution of this work.

Three design principles govern every stage of the system. *Provenance*: every extracted item is anchored to a verbatim quote from the source text, making the extraction verifiable against the original document. *Transparency*: each processing step produces independently inspectable intermediate outputs, and pipeline metadata is surfaced directly in the dashboard interface. *Traceability*: the system maintains a complete audit trail from raw input to structured record, so that any extracted claim can be traced back through the processing chain to its source paragraph. These properties are fundamental because the outputs are intended to inform policy decisions. An extracted claim that cannot be traced back to its source text provides no reliable basis for regulatory or legal reasoning [4, 5].

This paper describes the system and its application to the DFA consultation dataset, focusing on the pipeline architecture, interface design and practical lessons from processing a large and heterogeneous consultation dataset. Section 3 describes the dashboard and its main features. The paper is organised as follows. Section 2 discusses related work. Section 4 describes the pipeline in detail. Section 5 discusses the limitations of the system and key takeaways from the design and development process. Section 6 presents our conclusions.

2. Related Work

The legal NLP field has produced a substantial set of resources and benchmarks for EU regulatory text, including EURLEX57K [6], LEGAL-BERT [7], and the LexGLUE benchmark [8], establishing that EU legislative documents have distinct linguistic properties that reward domain-specific modelling. Apart from laws themselves as a source of data, more recently, Goanta et al. [9] argued that legal NLP should also systematically analyse regulatory processes. This framing is adopted in our work directly: we extend the RegNLP scope from analysis of enacted regulation to structured extraction from the upstream consultation submissions that inform it. Existing legal NLP overwhelmingly targets legislation, case law or contracts [10, 11, 12], generally leaving consultation submissions largely unexplored.

One of the earliest large-scale attempts to analyse regulatory submissions computationally is Livermore et al. [2], who applied topic modelling and duplicate detection to nearly three million US federal regulatory comments. Romberg and Escher [13] surveys subsequent work across multiple jurisdictions, finding that most systems rely on traditional NLP methods and explicitly calling for tools that keep human evaluators in the loop, a gap our auditability-centred design directly addresses. The closest prior work to ours is Di Porto et al. [3], who applied TF-IDF and BERT-based clustering to 830 consultation responses on the AI Act, DMA, and DSA from the same "Have Your Say" platform. Their approach generates document-level topic clusters, but no structured evidence records and no verbatim grounding. An earlier study applied simpler text analysis to DSA/DMA submissions [14]. Among more recent systems, Chenene et al. [15] combine actor detection, topic linking and RAG-based argument extraction for stakeholder debates, though they target media sources rather than formal regulatory submissions and lack verbatim grounding.

Argument mining and evidential reasoning provide the broader context for what structured evidence

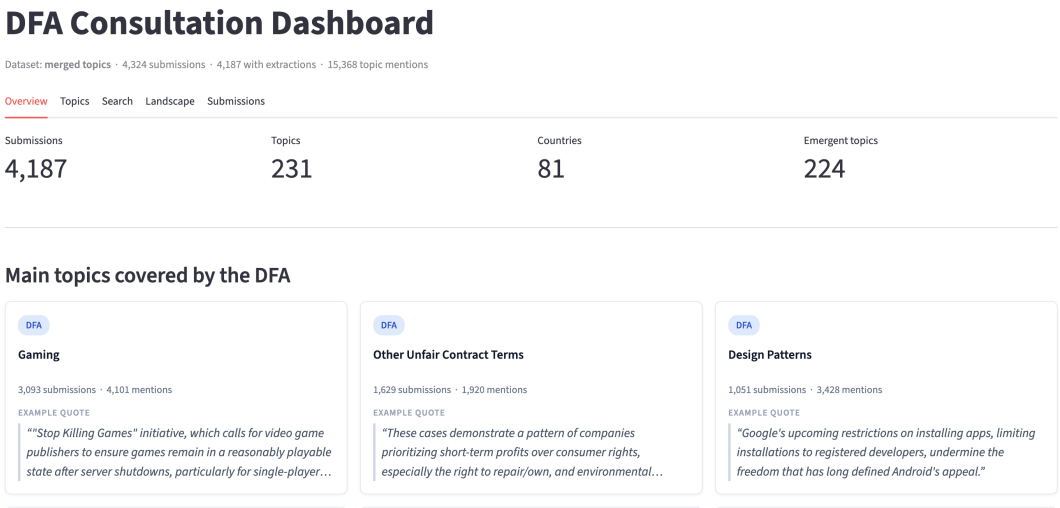


Figure 1: Overview tab of the DFA Consultation Dashboard. Headline metrics (submissions, topics, countries, emergent topic count) are shown at the top. Topic cards are split into predefined DFA categories (blue) and emergent topics identified by the model (orange). Each card shows submission count, mention count, and a representative anchor quote.

extraction should ultimately support. Lawrence and Reed [16] survey argument component identification across domains; Habernal et al. [17] extend this to legal settings with the largest annotated dataset for ECtHR proceedings. For evidential reasoning specifically, Bex et al. [4] establish that structured, traceable evidence is foundational to formal reasoning over legal claims. Our pipeline outputs – topic, stakeholder, verbatim quote and source paragraph – constitute the evidence layer on which this kind of downstream reasoning could operate.

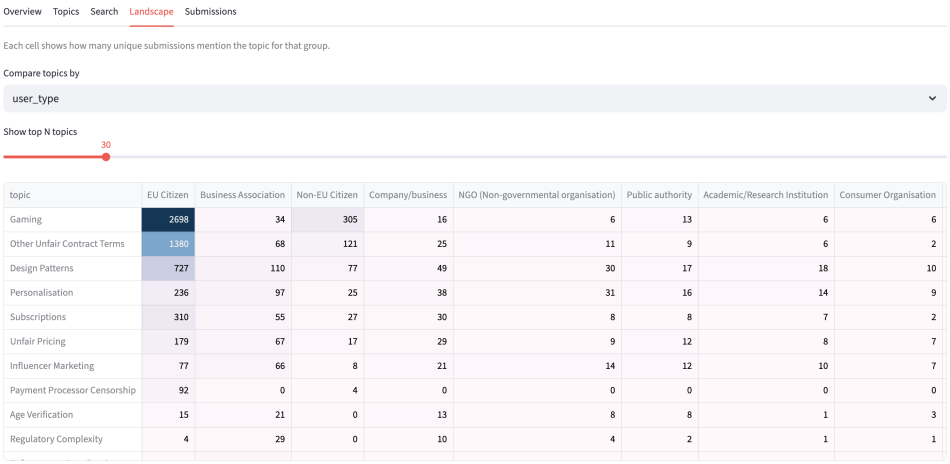
3. Dashboard

The dashboard is a web application publicly accessible at https://dfa-dashboard.thalesbertaglia.com/. It is organised around a persistent sidebar and five analytical tabs. All views respond to sidebar filter changes in real time, so every metric and evidence feed reflects the current filtered selection.

Sidebar: Filters and Provenance The sidebar serves two functions. The first is filtering: seven dimensions are available, stakeholder type, organisation name (free-text substring search), scope, governance level, company size, country, and language. All filters operate conjunctively and propagate to every tab simultaneously. The second function is provenance disclosure. A collapsible expander surfaces extraction run metadata read directly from the pipeline JSON outputs: model name, schema version, temperature, maximum tokens, and run start and finish timestamps. The sidebar also exposes a dataset version selector. Multiple labelled extraction runs can coexist in the outputs directory, each identified by a `label` file. Switching between runs updates all tabs instantly, enabling direct comparison across model versions or parameter settings.

Tabs The *Overview* tab provides a dataset-level summary of the extraction results. Figure 1 shows a partial view of the tab for the full DFA dataset. Four headline metrics are displayed at the top: total submissions, distinct topics, countries represented, and emergent topic count.

Below the metrics, topic cards are split into two groups. The first group covers the seven predefined DFA topic categories; the second covers emergent topics identified by the model. Each card shows the topic name, submission count, mention count, and an anchor evidence quote. DFA-registry cards are rendered in blue; emergent topic cards in orange. This colour coding is consistent across all tabs and



topic	EU Citizen	Business Association	Non-EU Citizen	Company/business	NGO (Non-governmental organisation)	Public authority	Academic/Research Institution	Consumer Organisation
Gaming	2698	34	305	16	6	13	6	6
Other Unfair Contract Terms	1380	68	121	25	11	9	6	2
Design Patterns	727	110	77	49	30	17	18	10
Personalisation	236	97	25	38	31	16	14	9
Subscriptions	310	55	27	30	8	8	7	2
Unfair Pricing	179	67	17	29	9	12	8	7
Influencer Marketing	77	66	8	21	14	12	10	7
Payment Processor Censorship	92	0	4	0	0	0	0	0
Age Verification	15	21	0	13	8	8	1	3
Regulatory Complexity	4	29	0	10	4	2	1	1

Figure 2: Landscape tab configured with stakeholder type as the column dimension.

provides the reader with an immediate visual signal about whether a topic was anticipated by the DFA's original framing or surfaced by the dataset.

The *Topics* tab shifts from a dataset-level overview to topic-level evidence. Selecting any topic from a dropdown reveals all source paragraphs that mention it, grouped by document. Each paragraph is rendered with its original text alongside the evidence quotes extracted from it, displayed in cards with submitter metadata badges. A right-hand column shows the stakeholder type breakdown as a bar chart and the top ten contributing countries, enabling immediate cross-stakeholder comparison for any single topic without leaving the view. The Search tab provides free-text search across all evidence quotes in the dataset, with case-insensitive substring matching. Each result is rendered as a quote card with its topic badge and stakeholder metadata. The primary use case is targeted retrieval: finding every submission that mentions a specific product name, company, regulatory instrument, or term.

The *Landscape* tab is the dashboard's main feature for policy analysis. Figure 2 shows the tab configured with stakeholder type as the column dimension. The view presents a topic-by-dimension pivot table: rows are topics (the top N by submission count, configurable via a slider with a default of 30), columns are any of six dimensions (stakeholder type, country, governance level, company size, language, or scope), and cell values are unique submission counts rendered with a heatmap colour scale. The dimension selector makes the same extraction data re-addressable from multiple analytical angles without any reprocessing. From any cell, the user can focus on the evidence quotes from exactly those submissions for that topic-dimension intersection. Selecting, for example, the cell at *Design Patterns* × *Business Association* surfaces the passages from business association submissions that discuss design patterns.

Finally, the *Submissions* tab allows users to view any individual submission in full, along with its extracted evidence. A searchable dropdown lists all submissions that match the current filters, each labelled by document ID, organisation, country, language, and date.

4. System Overview

The system transforms raw consultation submissions into structured, source-grounded evidence records and exposes the resulting data through an interactive dashboard designed for policy analysis. Figure 3 shows the end-to-end architecture. Two input paths feed a shared processing pipeline: PDF attachments are converted via OCR and segmented into paragraph-level units, while web-form feedback texts are ingested directly as single units. Both paths produce chunk JSON files that feed the LLM extraction stage. Extraction outputs are then consolidated with their source text and loaded into the dashboard.

Three principles govern every design decision in the system. *Verbatim grounding*: no paraphrase or summary is introduced at any stage; every extracted topic is anchored to a verbatim quote copied

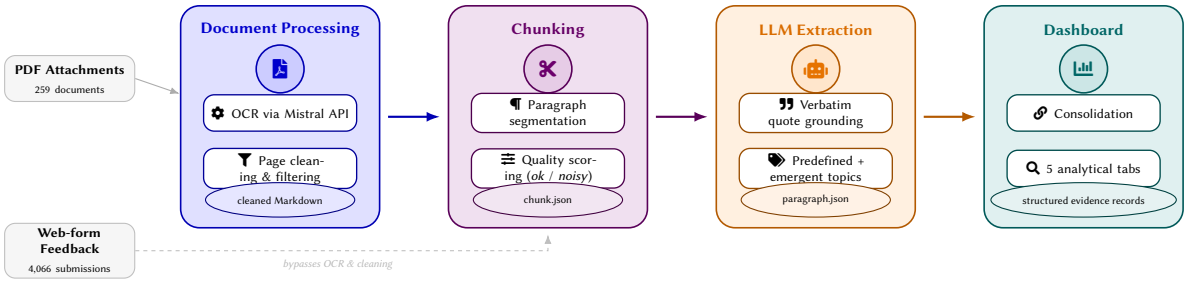


Figure 3: Pipeline architecture. PDF attachments (259 documents) and web-form feedback texts (4,066 submissions) follow distinct ingestion paths. PDFs are converted via OCR, cleaned page-by-page, and segmented into paragraph-level units. Feedback texts bypass these stages and are ingested directly as single units. Both paths converge at the chunking stage, where quality labels (*ok* / *noisy*) gate LLM access. The extraction stage produces verbatim-grounded topic annotations, which are consolidated with source text and surfaced through the interactive dashboard.

exactly from the source text. *Full traceability*: every item visible in the dashboard has an unbroken chain of evidence from the displayed evidence quote back to the source paragraph, source document and original submission. *Transparency by design*: no stage overwrites its input; every intermediate output is independently inspectable, and pipeline metadata is surfaced directly in the dashboard interface so that any result can be traced to the exact configuration that produced it.

The system is designed to be domain-generic. The OCR, cleaning, chunking, and LLM extraction components operate on any PDF or text corpus. The only domain-specific element is the topic schema embedded in the extraction prompt, which comprises seven predefined DFA issue categories with mapping instructions. Replacing this schema with another requires modifying a single prompt file. The DFA Consultation is one case-study of how the architecture can be used.

4.1. Document Processing Pipeline

The document processing pipeline transforms raw source documents into paragraph-level units ready for LLM extraction. Four stages handle ingestion, optical character recognition (OCR, to extract text from the PDF attachments), cleaning and filtering and chunking with quality scoring. Each stage writes named intermediate outputs rather than passing data in memory, so any stage can be re-run, inspected, or replaced independently, directly implementing our transparency principle.

Dataset The source dataset consists of submissions to the European Commission's "Have Your Say" platform for the DFA consultation, received between July and October 2025. The full submission set comprises 4,325 entries, of which 3 were excluded for being unavailable or empty. Of the remaining, 338 included PDF attachments (position papers and structured responses from institutional stakeholders) and the rest were text-only web-form responses. The PDF attachments were manually filtered to excluded non-submission files such as appended academic papers or annex-only documents, resulting in a final set of 259 documents. These two input types are handled differently downstream, as described below.

OCR PDF conversion is performed using the Mistral OCR API[1], selected for its handling of multi-column layouts and image-heavy documents that cause reading-order failures in rule-based tools. The API produces two outputs per document: a structured JSON representation and a per-page Markdown file. Both are retained as intermediate outputs alongside a `report.json` summarising conversion quality across the batch.

[1] https://mistral.ai/news/mistral-ocr

Cleaning and Filtering Cleaning operates on the OCR JSON output page by page. A sequential filter pipeline removes image placeholders, Markdown footnote definitions, isolated page numbers, and Unicode footnote markers, then normalises whitespace. Beyond these standard artefact removals, the pipeline applies three additional quality checks. First, automatic header and footer detection: lines that appear in more than 60% of pages across a document are identified as boilerplate and stripped. Second, lexical diversity checking: pages where the ratio of unique to total words falls below 0.15 are flagged as likely garbled OCR output and excluded. Third, table-of-contents detection: pages matching structural TOC patterns are removed as non-substantive. Pages falling below a 30-word threshold after cleaning are also excluded. Documents are then bucketed based on the number of pages dropped: documents losing two or fewer pages are written to `reliable/`; those losing more are written to `flag/`. Flagged documents are still processed downstream, but the flag indicates that extraction coverage should be interpreted with caution.

Chunking and Quality Scoring The goal of chunking is to segment cleaned documents into semantically coherent units sufficient for LLM extraction, which provide enough context to support verbatim quote extraction without diluting the topical signal. We parse cleaned Markdown using `markdown-it-py`[2], extracting paragraphs and headings, and list items with their section context preserved. Two input types are handled differently at this stage, distinguished by the `unit_kind` field. `paragraph` units are produced by the Markdown parser from PDF-derived documents and go through quality scoring. `feedback` units are the raw web-form text from the dataset CSV, ingested as a single self-contained unit per submission and assigned `chunk_quality="ok"` directly, bypassing scoring entirely. This reflects the nature of web-form responses: they are already the minimal meaningful unit and require no further segmentation.

For `paragraph` units, each chunk receives a meaning score: $\text{score} = 0.55 \times \ell + 0.45 \times \alpha - 0.35 \times \delta$ where ℓ is a length component, α an alpha-character ratio component, and δ a digit penalty, all clamped to $[0, 1]$. A hard floor applies: paragraphs with an alpha-character ratio below 0.35 receive a score of 0.0 regardless of other components. Chunks scoring at or above the default threshold of 0.50 are labelled `chunk_quality="ok"`; a top-k fallback (default $k = 60$) ensures minimum coverage even for short documents. Chunks labelled "`noisy`" are skipped by the LLM but recorded in the output, thus preserving a full account of what the model did not see. The output of this stage is a `<doc_id>.chunk.json` file per document containing paragraph text, section path, meaning score, selection flag and `unit_kind` for every chunk. This file is the primary auditing artefact for the extraction stage, as it allows a user to verify exactly which paragraphs of a given document were and were not presented to the model.

4.2. LLM-Based Extraction

LLM calls are managed via LiteLLM[3], which abstracts over model providers and makes the pipeline model-agnostic. We use *gpt-5-nano* with temperature 1.0. Only chunks with `chunk_quality="ok"` are sent to the model; skipped chunks are recorded with a `skip_reason` field.

Prompt Design The extraction prompt includes the paragraph text as its only variable input. The DFA topic schema (seven predefined categories with mapping instructions) is embedded directly in the prompt, along with explicit rules about extraction behaviour. The prompt instructs the model to extract only topics explicitly supported by the text, not to infer or generalise beyond what is written, and to return empty output for passages consisting of names, job titles, affiliations, signatures, headers, or procedural text. We hypothesise that these negative rules contribute to the 33.6% empty-output rate across chunks sent to the model.

The verbatim grounding rules are also included in the prompt: every extracted topic must include at least one evidence quote copied exactly from the passage, with no paraphrasing, no shortening, no

[2] https://github.com/executablebooks/markdown-it-py
[3] https://litellm.ai

normalisation, and no correction. The prompt specifies the smallest exact quote that clearly supports the topic, which may be a full sentence or a supporting fragment, but must appear verbatim in the source. We enforce the implementation of the provenance principle using a Pydantic schema requiring at least one non-empty string in the `evidence_quotes` field, thus ensuring that the models always outputs a quote. We verified verbatim grounding by matching all extracted quotes against their source paragraphs using exact and fuzzy string matching, achieving an overall match rate of 99.1%.

Each extracted item carries a `topic_source` flag: "`dfa`" for topics mapped to one of the seven predefined DFA categories, and "`new`" for topics the model identifies outside that schema. The prompt instructs the model to prefer predefined labels when they fit and to use new labels only when no predefined category reasonably applies, and to make new labels short, specific and concrete rather than generic abstractions. This hybrid design ensures coverage of the issues the DFA was designed to address while remaining open to concerns stakeholders raise beyond that framing. In the DFA run, 88.6% of extracted items mapped to predefined labels; 11.4% were emergent, producing 1,458 unique raw emergent labels. Many of the generated emergent topics are near-duplicate or redundant labels, so we perform a post-processing step to increase their quality.

Topic Post-Processing The 1,458 unique emergent labels produced by the raw extraction run require consolidation before they are analytically usable. A three-step post-processing pipeline addresses this. First, emergent labels are mapped to the seven predefined DFA topics using semantic similarity, and unmatched labels are passed to the next step. Second, remaining labels are grouped via agglomerative clustering; each cluster receives a canonical label selected by frequency and brevity. Third, residual low-frequency labels are matched to the nearest existing group via similarity and reviewed manually before final assignment. This procedure reduces the emergent label space from 1,458 to 224 distinct topics while preserving substantively meaningful niche concerns that would otherwise be lost. The dashboard exposes both the raw and post-processed topic sets, letting analysts inspect the effect of consolidation directly.

Finally, the consolidation step re-joins LLM extraction records with their source paragraph text from the chunk files. For each extracted paragraph record, the script locates the matching chunk by `para_id` and attaches the original text alongside optional chunk metadata. The final consolidated record carries: `doc_id`, `para_id`, `topic`, `topic_source`, `evidence_quotes`, and the source paragraph text. This results in a complete traceability chain – from any item in the dashboard, a user can follow this chain back to the exact text the model was shown.

5. Limitations and Practical Considerations

Building and deploying the pipeline on a real consultation dataset surfaces observations not visible from the architecture alone. This section documents what we found in practice: where the design held up, where it created friction, and what we would recommend to anyone applying a similar approach.

Transparency as a design requirement Every pipeline parameter shapes what gets extracted: the meaning score threshold determines which paragraphs the model sees; the topic schema determines which issues are predefined versus emergent; the prompt's negative rules determine what counts as substantive content. These are interpretive choices with real consequences in a policy context. Our recommendation is to treat transparency as a first-class requirement from the start. Concretely: retain intermediate outputs at every stage, version the prompt alongside the code, and surface pipeline configuration directly in any interface that presents results. The dashboard's provenance expander operationalises this principle.

OCR and chunking are the real bottlenecks LLM extraction quality is bounded by text quality, which is set upstream by OCR and chunking. These steps receive far less attention than prompt engineering in most LLM pipeline discussions, but in practice, they are where the most significant

and least recoverable errors occur. OCR failures, such as reading-order errors in multi-column layouts, garbled tables, misidentified page boundaries, corrupt the text units that all subsequent steps operate on. No prompt refinement recovers a position paper whose columns were read across rather than down. Manual sampling of a representative subset of documents before running the full pipeline is worth the time. The `flag/` bucketing mechanism surfaces problematic documents but does not fix them. Chunking introduces a second layer of loss: the meaning score heuristic handles standard narrative prose well but degrades on tables, itemised policy recommendations, and short formulaic statements that carry substantive content. We recommend treating the threshold as a tunable parameter and spot-checking excluded chunks on a sample of documents. LLM-assisted segmentation by content structure rather than typographic convention is the most promising longer-term improvement.

Emergent labels: a useful tradeoff with a real cost The raw extraction produced 1,458 unique emergent topic labels, consolidated to 224 after post-processing. Near-duplicate labels(such as *Payment Processor Censorship*, *Payment Processor Gatekeeping*, *Payment Processor Pressure*) are a direct consequence of processing each paragraph independently without a shared vocabulary across documents. This is a known tradeoff: the same independence that generates near-duplicates also surfaces genuine niche concerns that a more constrained approach would suppress, including policy signals such as Age Verification, Digital Ownership, and Enforcement Coordination that a fixed taxonomy would miss entirely. The practical implication is to plan for post-processing from the start, treat the cluster distance threshold as a parameter requiring calibration per dataset, and build a manual review step into the workflow for low-frequency labels. The dashboard exposes both raw and merged topic sets so users can inspect the consolidation rather than accept it uncritically.

Verbatim grounding The verbatim quote constraint is enforced through prompting and schema validation. Post-hoc verification via exact and fuzzy string matching against source paragraphs confirmed a 99.1% match rate across all extracted quotes. There are still some quotes that do not match the source text, indicating potential model hallucinations that should be further analysed.

6. Conclusion

We presented an end-to-end NLP pipeline and interactive dashboard for structured evidence extraction from regulatory public consultation submissions, applied to the EU Digital Fairness Act dataset as a case study. The system processed 4,322 submissions and produced 15,368 topic annotations, grounded in 20,951 verbatim evidence quotes. Three principles govern the design throughout: verbatim grounding, full traceability, and transparency by design.

The DFA application demonstrates the system's practical value. The hybrid predefined-plus-emergent topic schema surfaced policy signals outside the DFA's original framing (including Age Verification, Payment Processor Censorship, and Digital Ownership) that a fixed-taxonomy approach would have missed entirely. The Landscape tab makes cross-stakeholder analysis tractable at scale while preserving a direct path from aggregate pattern to verbatim source text. The provenance expander ensures every result is traceable to a specific, reproducible pipeline configuration.

The pipeline is domain-generic and designed for reuse. Adapting it to a new consultation requires updating the extraction prompt and supplying new data; the pipeline architecture, dashboard, and output format require no modification. As regulatory consultations grow in volume and policy significance across EU digital governance, tools that make large-scale stakeholder testimony systematically accessible while remaining faithful to source material are increasingly necessary. This system is a concrete step in that direction.

Future work include the evaluation of other models (such as open-source LLMs), LLM-assisted chunking by argumentative unit, analysis of the emergent topic consolidation procedure across multiple consultation datasets, and systematic evaluation of the quality of the extracted topics. Pipeline code

and DFA extraction outputs are publicly available at https://github.com/thalesbertaglia/dfa-dashboard to enable replication and extension.

Acknowledgments

This research has been supported by funding from the ERC Starting Grant HUMANads (ERC-2021-StG No 101041824).

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT to check grammar and improve the writing in some sections. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] European Commission, Call for evidence for an impact assessment - ares(2025)5829481, 2025.
[2] M. A. Livermore, V. Eidelman, B. Grom, Computationally assisted regulatory participation, Notre Dame L. Rev. 93 (2017) 977.
[3] F. Di Porto, P. Fantozzi, M. Naldi, N. Rangone, Mining eu consultations through ai: Fd porto et al., Artificial Intelligence and Law (2024) 1–38.
[4] F. J. Bex, P. J. Van Koppen, H. Prakken, B. Verheij, A hybrid formal theory of arguments, stories and criminal evidence, Artificial Intelligence and Law 18 (2010) 123–152.
[5] H. Rashkin, V. Nikolaev, M. Lamm, L. Aroyo, M. Collins, D. Das, S. Petrov, G. S. Tomar, I. Turc, D. Reitter, Measuring attribution in natural language generation models, Computational Linguistics 49 (2023) 777–840. URL: https://aclanthology.org/2023.cl-4.2/. doi:10.1162/coli_a_00486.
[6] I. Chalkidis, E. Fergadiotis, P. Malakasiotis, N. Aletras, I. Androutsopoulos, Extreme multi-label legal text classification: A case study in EU legislation, in: N. Aletras, E. Ash, L. Barrett, D. Chen, A. Meyers, D. Preotiuc-Pietro, D. Rosenberg, A. Stent (Eds.), Proceedings of the Natural Legal Language Processing Workshop 2019, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 78–87. URL: https://aclanthology.org/W19-2209/. doi:10.18653/v1/W19-2209.
[7] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, I. Androutsopoulos, Legal-bert: The muppets straight out of law school, in: Findings of the association for computational linguistics: EMNLP 2020, 2020, pp. 2898–2904.
[8] I. Chalkidis, A. Jana, D. Hartung, M. Bommarito, I. Androutsopoulos, D. Katz, N. 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), 2022, pp. 4310–4330.
[9] C. Goanta, N. Aletras, I. Chalkidis, S. Ranchordás, G. Spanakis, Regulation and NLP (RegNLP): Taming large language models, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 8712–8724. URL: https://aclanthology.org/2023.emnlp-main.539/. doi:10.18653/v1/2023.emnlp-main.539.
[10] H. Surden, Machine learning and law, Washington Law Review 89 (2014) 87–115.
[11] H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu, M. Sun, How does nlp benefit legal system: A summary of legal artificial intelligence, in: Proceedings of ACL 2020, 2020, pp. 5218–5230.
[12] M. Siino, M. Falco, D. Croce, P. Rosso, Exploring LLMs applications in Law: A literature review on current legal NLP approaches, IEEE Access 13 (2025) 18253–18276.
[13] J. Romberg, T. Escher, Making sense of citizens' input through artificial intelligence: A review of methods for computational text analysis to support the evaluation of contributions in public participation, Digital Government: Research and Practice 5 (2024) 1–30.

[14] F. Di Porto, T. Grote, G. Volpi, R. Ivernizzi, "i see something you don't see": A computational analysis of the digital services act and the digital markets act, Stan. Computational Antitrust 1 (2021) 84.
[15] M. Chenene, J. Rouhier, J. Daniélou, M. Sarkar, E. Cabrio, Stakeholder suite: A unified ai framework for mapping actors, topics and arguments in public debates, in: Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 3: System Demonstrations), 2026, pp. 1–20.
[16] J. Lawrence, C. Reed, Argument mining: A survey, Computational linguistics 45 (2019) 765–818.
[17] I. Habernal, D. Faber, N. Recchia, S. Bretthauer, I. Gurevych, I. Spiecker genannt Döhmann, C. Burchard, Mining legal arguments in court decisions, Artificial Intelligence and Law 32 (2024) 1–38.