

AI as a Research Assistant for Regulatory Studies – a Methodological Note

In brief...

A brief guide to using large language models as research assistants to study public participation in rulemaking (and, more broadly, regulation as a whole).

By: Lucas Thevenard | February 16, 2026

Regulation is built, justified, contested, and revised through text: draft rules, technical analyses, public comments, and agency responses. That written record is a gold mine for scholars—but it is also a practical barrier. When the evidence is mostly presented in the form of text documents, measurement is slow, difficult to standardize, and hard to scale across many proceedings or years.

In my PhD research, I studied responsiveness to public input by ANATEL (Brazil's telecommunications regulator), using thousands of consultation submissions linked to the agency's responses, which indicate whether each suggestion was accepted, partially accepted, rejected, or treated as out of scope. Doing that work forced me to confront three recurring research challenges: identifying who is in the record, measuring what participants are asking for and how they justify it, and connecting that content to agency responses in a meaningful, rigorous way.

This methods note tells that story, emphasizing a methodological tool that can greatly improve and expand efforts to study regulation: using a general-purpose AI model as a research assistant. While acknowledging multiple AI-based approaches to text analysis (text representations, predictive modeling, and training specialized models), in this piece I focus primarily on the most intuitive workflow—one that can be useful for a wide variety of research applications and research teams: LLM-assisted coding, that is, using general-purpose AI models to read text and extract specific, auditable measures that can be validated through traditional methods and used in social science research.

1. Why use AI models to study regulation?

Modern regulation is text-intensive. In the United States, rulemaking under the Administrative Procedure Act generates long Federal Register notices, dense preambles that explain agency reasoning, supporting

analyses (including regulatory impact analyses), public comments, and agency responses. This is a rich empirical record, but it is also an inconvenient one: the core evidence for understanding regulation is often only available as prose.

In practice, three constraints show up again and again.

The first is **scale**. The sheer amount of text to be read, processed, or understood tends to push researchers toward simplification. A single docket can include thousands (sometimes tens of thousands) of pages of text, so fine-grained measurement becomes difficult to sustain across many rules, agencies, or years. This is why much of the empirical notice-and-comment literature focuses on a single agency or a small set of rules; yet the resulting evidence is often treated as if it establishes general truths about regulation as a whole, without systematic tests of cross-agency variation.

The second is **heterogeneity**. Regulatory writing comes in many genres—legalistic, technical, economic, rhetorical—and those genres vary across agencies, sectors, and time. Measures that rely on a fixed dictionary of terms can be fragile when the same idea can be expressed in multiple ways (or when the same word means different things in different contexts). The solution most often used by social science scholars is to rely on qualitative close reading of the texts, but again this runs into the scale problem discussed above.

The third is **linking documents to outcomes**. Many of the most interesting questions in regulatory studies require connecting texts that are naturally connected in the process but hard to connect analytically: proposed to final language, comment to response, or argument to acceptance. Without scalable ways to extract content variables from each document, those links tend to be studied indirectly or in very small samples.

AI does not remove the need for judgment, theory, or careful research design. What it can do—when used conservatively and transparently—is reduce the marginal cost of reading at scale, allowing researchers to efficiently extract meaningful, structured information from the vast regulatory corpus. It makes it more realistic for small teams to treat parts of the written record not only as an archive but also as data.

Three ways to use AI for text analysis

There are multiple ways to “use AI” in text-based research. In my ANATEL project, I used all three approaches below at different stages. These approaches serve different purposes and require different kinds of technical investment.

1. **Statistical text representations (e.g., TF-IDF and embeddings)**. These approaches convert each document into a numeric representation that summarizes patterns of word use, linguistic similarity, or even semantic content. They are useful for tasks like clustering, measuring similarity, reducing dimensionality, or building predictive models of outcomes. They can be powerful—but the representations themselves are not “variables” in the usual social science sense unless researchers do additional work to interpret and validate what they capture.
2. **Training a specialized classifier (supervised learning)**. Here the goal is to build a model tailored to a specific label of interest (for example, whether a comment is within scope, or whether it will be accepted). This can work well when a research team has a labeled training set and wants consistent, repeatable classification at scale. The main cost, however, is that it requires creating a large enough training dataset and maintaining the modeling workflow.

3. **LLM-assisted coding (using AI as a research assistant).** This approach treats a general-purpose large language model (LLM) as a constrained reader. It requires researchers to define the construct they care about, specify the decision rules (much like a codebook), and ask the model to extract a small set of structured outputs (for example, a few categories or numeric codes). Crucially, researchers will then validate those outputs by auditing samples, revising the instructions when needed, and keeping the prompts and logs so others can inspect what has been done.

This piece focuses primarily on the third approach (LLM-assisted coding), because it is often the most intuitive entry point for non-technical research teams. It can be implemented with relatively lightweight tooling, produces measures that look like familiar research variables, and can be made auditable with straightforward validation routines. The rest of the note shows how I used this “AI research assistant” workflow to overcome research challenges while exploring the three most common questions social science researchers like to pose about public participation and agency responsiveness.

2. Three questions, one powerful ally: how I used the “AI assistant” in my own research

When I began my dissertation project on ANATEL’s public consultations, I knew I would have to tackle some of the most common questions in this type of research: *who participates?*, *what do the comments say?*, and *how does the agency respond?* Those questions are the backbone of much notice-and-comment scholarship. But once I started working with the raw record, it became clear that each one hides a practical challenge: the information you want is often present somewhere in the documents, yet it is rarely available in a clean, structured form that can be analyzed at scale.

Who participates?

I started where most researchers start: with the participation metadata. ANATEL’s consultation spreadsheets include fields for the *author* of a submission and, sometimes, the *entity* the author represents. In principle, that should make it straightforward to categorize commenters into interest groups such as *private companies*, *trade associations*, *government bodies*, *consumer groups*, and so on.

However, in practice those fields were not enough. Many entries were incomplete (an individual name, no affiliation). Others were inconsistent (the same organization written in multiple ways, acronyms without expansion, and the like). And, importantly, the category that ends up swallowing all ambiguity is the easiest one to create and the hardest one to interpret: “Individual.” In my early versions of the dataset, “Individual” was not a meaningful identity category. It was a residual category, a placeholder for “the record does not tell who this person is.”

That residual category quickly became a problem rather than a neutral choice. If half the dataset sits in a bucket labeled “Individual,” it becomes difficult to say anything serious about participation patterns. Are individual citizens really dominating the process—or are many of these “individuals” actually sector professionals, radio amateurs, or representatives of firms who simply did not fill the metadata fields in a consistent way?

I could have tackled this the traditional way: read the comments one by one and look for identity cues in the text (“I write on behalf of...,” “as an engineer...,” “as a union representative...,” “as a company operating in...,” and so forth). But that would have meant close reading almost six thousand submissions just to recover basic information about who was in the record, before even reaching the substantive questions about what they asked for and how the agency responded.

This was the first place I began treating a large language model as a research assistant, to scale a familiar research task: *finding relevant evidence in text*.

Concretely, I had the model go through the cases that were initially classified as “Individual” and ask a narrow, conservative question: Is there explicit evidence in the comment that the author belongs to, or is writing on behalf of, a specific interest group? If the answer was yes, the model had to (1) propose the interest-group category and (2) quote or point to the specific textual evidence that justified that proposal. If the answer was no—or if the text was ambiguous—the model was instructed to leave the case as “Individual” rather than guess.

That “evidence-first” design mattered. The output I wanted was not a black-box relabeling of the dataset. It was a triage tool: a way to surface the subset of comments where the text itself contained strong identity signals, along with the passages I would need to verify them.

The workflow then became straightforward:

- The model scanned thousands of residual “Individual” submissions and produced a list of *candidates* where it found explicit affiliation evidence.
- For each candidate, it returned the proposed group label *and the snippet of text* that supported the label.
- I manually reviewed those cases—accepting the classification when the evidence was clear, rejecting it when it was weak, and keeping the audit trail either way.
- I also audited a small sample of cases where the model had not found any evidence of identity signals in the text, reading those comments fully to ensure that the model was not excessively conservative and systematically missing strong cues that could reduce the residual category further.

This LLM-assisted “research assistant” pass was highly effective: it reduced the residual “Individual” category from 4,795 comments to 3,054—reclassifying 1,741 comments ($\approx 36\%$ of the residual) into substantively meaningful interest-group categories. The goal here was not just to achieve higher “accuracy” in an abstract sense but to shrink an amorphous residual category into a more meaningful map of participants—while keeping the researcher in the loop and minimizing the risk of injecting spurious identity information into the dataset.

What do the comments say?

Once I had a clearer map of *who* was in the record, the next question was the one that motivated the project in the first place: *what are participants actually asking for—and how do they justify those requests?* My goal was not simply to describe comment topics. It was to understand the interplay between **interest groups** and **ideas**: whether different kinds of participants systematically advance different kinds of claims, and whether certain kinds of reasoning appear more (or less) aligned with the agency’s decision-making.

This is where traditional qualitative content analysis runs into its hardest constraint. If you want to read deeply, you can learn a great deal from a few consultations. But once you want to compare patterns across hundreds of proceedings, you need content measures that are consistent, auditable, and scalable.

In my project, I used the “AI research assistant” workflow to create two theory-aligned variables from the text of each comment: (1) the **direction of requested regulatory change** and (2) the **types of arguments** used to justify that request.

Direction of requested regulatory change

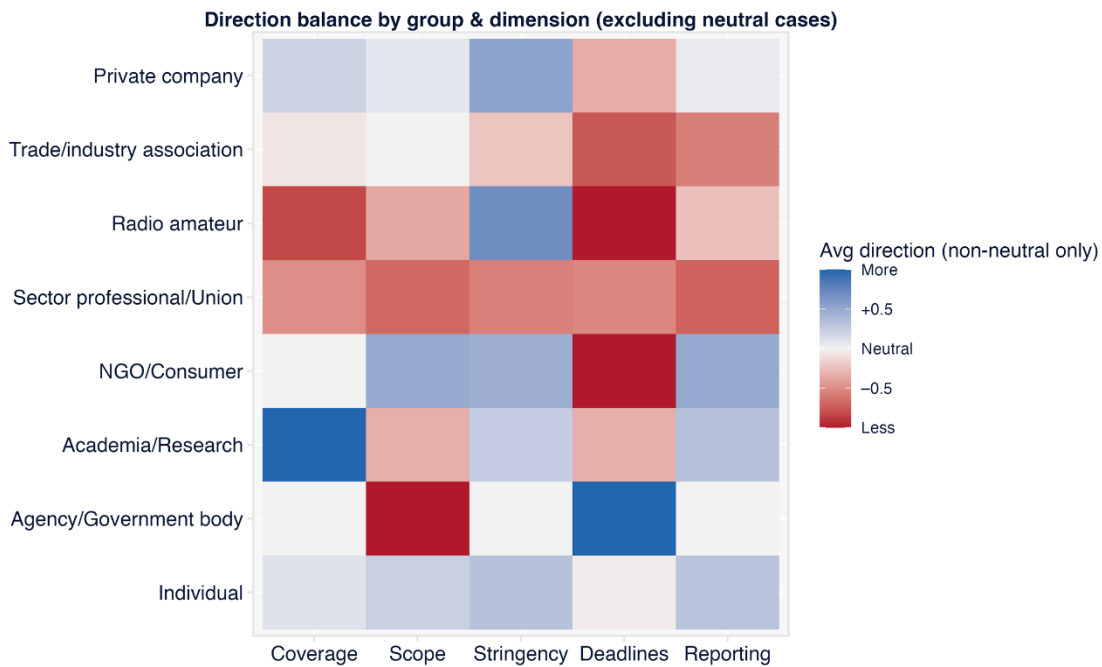
One of the simplest ways to describe participation is to ask whether commenters want *more* or *less* regulation. But as soon as you start reading actual submissions, the question becomes slippery. “More” can mean *more coverage* (more entities covered), *broader scope* (more outcomes or activities regulated), *higher stringency* (stricter standards), *shorter deadlines* (faster compliance), or *more monitoring and reporting*. Those are different levers—and participants can easily ask for “more” on one dimension and “less” on another.

To capture that complexity, I adapted a structured coding scheme developed by [Balla et al. \(2022\)](#) that classifies each comment on these five dimensions of regulatory design, using a simple directional code: -1 (less), 0 (neutral / not addressed), or +1 (more). I then asked the model to behave like a careful human coder: apply the definitions conservatively, and return 0 whenever the text was silent, ambiguous, or off-topic for a given dimension.

Practically, this looked like giving the model a short “codebook” (definitions + examples) and asking it to return a constrained, machine-readable output for each comment. Just as importantly, the workflow enabled accountability: I audited random samples of classifications for each of the dimensions, revised the instructions when disagreements revealed ambiguity, and kept the prompts and logs so the coding could be inspected and repeated.

The result is illustrated in Figure 1, which shows each group’s tendency (mean regulatory change score) for each dimension, considering only the cases where there was a clear preference.

Figure 1. Direction balance by group and regulatory dimension (excluding neutral cases; -1 = less regulation, +1 = more regulation).



One pattern I did not fully anticipate until I could measure it systematically was how sharply private companies and trade/industry associations diverged across regulatory levers. In this corpus, private companies leaned strongly pro-regulatory on *stringency*, while trade associations leaned deregulatory on *deadlines* and *reporting*.

One plausible explanation for this divergence is that many firm submissions come from large incumbents that already operate under high technical standards. Their comments often push to incorporate those standards into binding rules and to clarify or expand regulatory scope, both to reduce uncertainty and to level the playing field across the sector. Trade associations, by contrast, frequently represent smaller companies or more heterogeneous memberships (including actors from adjacent economic sectors). Those constituencies have stronger incentives to resist tighter technical requirements or broader regulatory reach and to press for longer implementation timelines and lighter reporting obligations.

That is exactly the kind of within-“business” heterogeneity that can be lost when interest-group categories are too coarse, and it offers a concrete way to connect participation research to more precise theories about industry structure, incumbency, and collective representation.

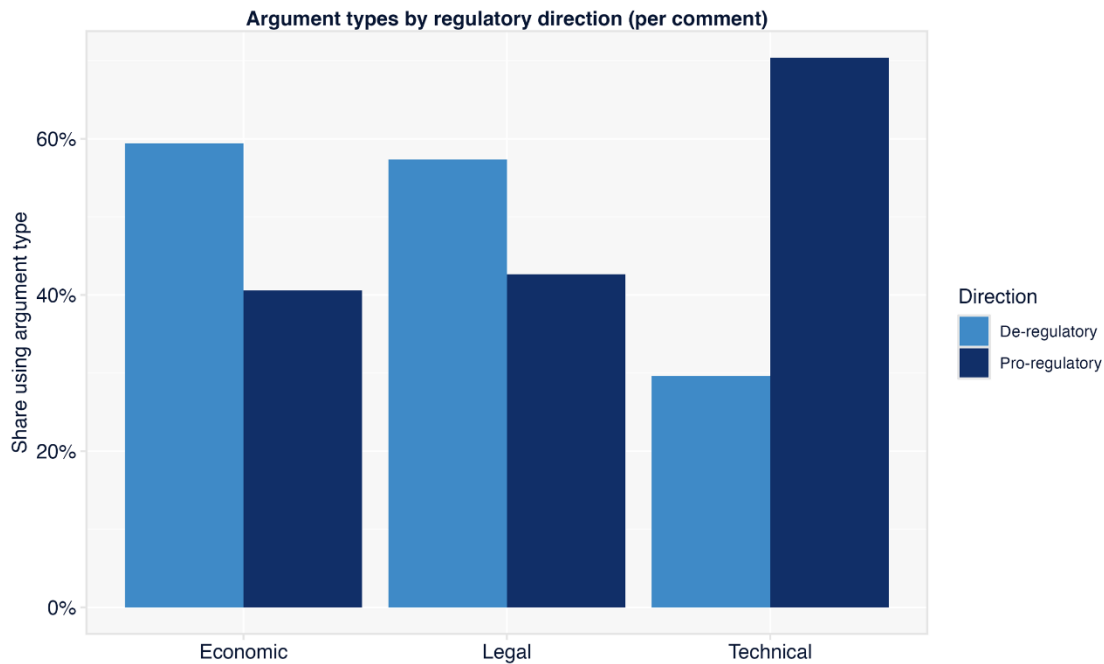
Types of argument (how commenters justify what they ask for)

Direction captures *what* a commenter is asking for. But for my purposes, it was just as important to capture *how* they justified the request. In a technically specialized agency like ANATEL, it is plausible that the form of reasoning matters: some submissions read like engineering memos; others lean on legal authority; others emphasize compliance costs, market structure, or economic feasibility.

To operationalize that idea, I used LLM-assisted coding to create three non-exclusive indicators, capturing whether a comment meaningfully relied on technical, legal, or economic reasoning. As before, the model’s task was deliberately narrow and conservative: it had to return simple binary flags (1/0) and code “0” when the argumentative signal was weak or unclear. I again audited samples to verify that the model was not hallucinating argument types into generic language.

Figure 2 compares argument types across comments that, overall, asked for more versus less regulation. Again, the point is not that the relationship is causal: argument types also correlate with policy area, feasibility, and the identities of participants. The point is that once these variables exist as auditable measures, we can finally test competing explanations about what kinds of comments are prevalent in notice-and-comment procedures, which tend to be most effective, among other long-standing questions.

Figure 2. Argument Types by Regulatory Direction (per comment)



The data reveal some interesting patterns. First, **technical reasoning** clustered strongly with pro-regulatory requests, consistent with the idea that many “more regulation” arguments in this setting take the form of engineering or performance claims. Second, **economic** and **legal** argumentation skewed more toward deregulatory requests, suggesting a different justificatory repertoire: costs, burdens, and authority constraints rather than technical performance. Those descriptive patterns mattered for my broader research intent, because they offered a concrete way to study the “ideas” side of participation—how different groups frame their claims, and which frames appear to travel together with different policy directions.

How does the agency respond?

The third question is often the hardest: *how do agencies respond to public input?* In ANATEL’s case, I was fortunate to have something that many U.S.-based notice-and-comment researchers do not: a labeled dataset in which the agency itself provides individualized, categorical responses to each comment, indicating whether a suggestion was accepted (fully or partially), rejected, or treated as out of scope.

That kind of comment-level labeling is not common in the U.S. context. In many rulemakings, researchers have to reconstruct responsiveness indirectly: tracking changes from proposed to final rule text, identifying which comments requested those changes, and distinguishing accepted requests from requests that were ignored or rejected. Each of those steps typically requires substantial coding—exactly the kind of work an “AI research assistant” can help with by extracting structured measures of what each comment is asking for and by making large-scale document comparison more feasible.

In my case, however, the challenge was different. Because the outcomes were already labeled, the question became: what can we learn by combining that labeled text with statistical text analysis tools? This is where the broader AI toolkit opens up. Using semantic representations of text (embeddings) and standard

supervised learning methods, we can build predictive models that learn patterns linking *what is said* in a comment to *how the agency responds*.

I cannot walk through the full modeling methodology here, since that would go far beyond the scope of a short methods note. Instead, Table 1 summarizes validation-stage performance in the impact classification task, in which I trained models (using GPT embeddings) to predict whether within-scope comments would be accepted or rejected. In other words, the table reports how well the models predict outcomes for “new” comments they did not see during training.

Table 1. Validation performance for the impact classification task (accepted vs. rejected), using GPT embeddings.

Algorithm	Accuracy (validation)	Macro-F1 (validation)
Support Vector Machines	83.4%	83.4%
K-Nearest Neighbors	81.8%	81.8%
Random Forest	78.7%	78.5%
Neural Net (MLPC)	78.7%	78.6%
AdaBoost	77.5%	77.5%

The surprising takeaway here is that the most accurate models could predict the agency’s response for most comments (83.4%, for the best model) even on the validation set—suggesting that ANATEL’s accept/reject decisions follow relatively stable, learnable patterns that can be extrapolated and predicted from the substantive content of the submissions.

For me, that predictive result immediately raised a new set of questions: why was responsiveness so predictable in the first place, and what were the most relevant predictors of the agency’s decisions? Those are the kinds of questions that predictive models cannot answer on their own, and that the explanatory models in my broader research could then attempt to untangle.

That is exactly the kind of empirical leverage that becomes possible once regulatory text is represented in ways that can be modeled—an extension of the AI toolkit for research teams willing to go beyond coding and into predictive and explanatory analysis.

3. Payoffs and drawbacks: The Promise and Limits of LLM-assisted Coding

In the previous section, I briefly stepped beyond LLM-assisted coding to show how statistical text representations can enable predictive (and eventually explanatory) modeling of agency responsiveness. But the main focus of this note is the most intuitive—and, for many teams, the most immediately useful—tool in the AI toolkit: treating a general-purpose model as a disciplined research assistant that helps turn text into auditable measures. Used carefully, the “AI research assistant” workflow can change what is feasible in text-based regulatory research.

In what follows, I highlight the main benefits and drawbacks of using this tool. In my experience, the benefits come in two main forms: resource benefits (what it does to the cost and timeline of a project) and methodological benefits (what it does to the quality, consistency, and transparency of measurement).

Resource benefits

The most immediate payoff is efficiency. A general-purpose model can read and extract structured information from large volumes of text in a short period of time. In practical terms, a single computer running this kind of workflow can often complete in hours what would otherwise require a large research team working for days.

In my own project, I initially planned to read all 5,814 comments—and the main supporting documents for 219 consultations—by myself. After more than 40 days of intense work, I had made it through roughly 80% of that reading load. At that point, I shifted to an AI-first measurement strategy: it took me two days to prepare and test the scripts, and five days to run the coding and audit the results. In about one week, I had reliable classifications for the full universe.

That efficiency gain does more than save time. It changes the way a project can be managed and designed. When measurement is no longer the overwhelming bottleneck, researchers can spend more of their effort on what the data are ultimately *for*: developing explanations, testing competing theoretical accounts, and making sense of patterns rather than just extracting them. It also makes it more realistic to pursue ambitious projects with smaller, more coherent research teams, instead of building large teams primarily to do manual coding.

Finally, these workflows are reusable. Once the coding logic is implemented in scripts—data cleaning steps, prompts, parsing routines, and auditing procedures—much of it can be adapted to expand the same project (more years, more consultations) or to apply a similar measurement strategy to another agency or regulatory setting.

Methodological benefits

Beyond speed, the “AI assistant” approach can improve the rigor of measurement by making it more standardized, more transparent, and easier to reproduce. Traditional close reading often relies on tacit judgment that is hard to fully document. By contrast, an LLM-assisted workflow tends to force researchers to record their decisions in explicit, inspectable form: the steps live as code, and the coding rules live as prompts that function like an operational codebook.

That structure also helps with consistency across time and across coders. Once prompts are written as clear decision rules, the same definitions can be applied uniformly across thousands of documents, instead of drifting as researchers get tired or as a team informally revises its interpretation of a category. And because the coding is scripted, it becomes much easier to share the materials—prompts, code, and logs—so that other researchers can understand what was done and, in principle, reproduce it.

Finally, this workflow naturally pairs coding with auditing. In practice, I treated each AI coding step as something like an inter-coder reliability exercise: I independently coded random samples of cases and compared my judgments to the model’s outputs, revising the prompts when disagreements revealed ambiguous definitions or predictable failure modes. Had I coded the full set of comments entirely on my own, it would have been difficult to offer any comparable reliability check without recruiting a second human coder. With AI-assisted coding, I was able to build validation directly into the measurement process.

Drawbacks

These payoffs are real, but they come with costs—and researchers should be clear-eyed about two of them.

Reproducibility in a fast-moving field. Large language models are improving quickly, and providers update or replace models over time. Even when the coding prompt stays the same, outputs can shift across model versions or settings, which can make it harder for other scholars to reproduce results years later. Fortunately, this is relatively straightforward to mitigate: treat the model as part of the method, and document it accordingly. In my own work, I used OpenAI’s platform and reported which model was used for which task, alongside the procedures and auditing steps. More generally, researchers can record the specific model/version, key settings, prompts, and logs—and, when feasible, rerun a subset of the coding with a second model to test whether the main patterns are robust rather than artifacts of a single system.

Distance from the object of study. The whole point of using an AI assistant is to reduce the reading burden—but relying on it too heavily can also distance researchers from the texts they are trying to understand. In my case, the weeks I spent reading comments before shifting to AI were not wasted; they gave me a deeper feel for the corpus, made me more confident in what the coding outputs meant, and helped me interpret the patterns I later measured. There is no simple technical fix here. The best safeguard is procedural: researchers should build in deliberate opportunities for immersion—reading a meaningful sample early on, doing targeted close reading when results are surprising, and treating auditing not as a box-checking exercise but as ongoing engagement with the record. Used well, AI disciplines human judgment and allows researchers to scale their projects; used carelessly, it can replace judgment and weaken understanding of what is being studied.

Conclusion

The core message of this note is simple: AI tools do not make regulatory research automatic, but they can make text analysis scalable and expand the ways we work with the written record. Used as a disciplined research assistant—with clear definitions, constrained outputs, and routine auditing—LLM-assisted coding, in particular, can help researchers turn regulatory text into structured evidence for addressing long-standing questions about participation, ideas, and responsiveness.

Looking ahead, the most important contribution of these tools may be what they make *possible* to study. When researchers can extract auditable measures from text at scale, they can move beyond small samples and one-off case studies toward more cumulative work: comparing participation across agencies and years, tracing how arguments and requests map onto outcomes, and building models of responsiveness that can be tested against competing theories. With careful documentation and routine auditing, AI-assisted workflows can help turn the vast regulatory record into a more accessible empirical resource—and open new pathways for rigorous research.

Annex: Example Prompts Used for LLM Coding

As discussed in this Insight, in my research I used structured prompts to convert comment text into replicable labels. Concretely, I implemented a Python pipeline that iterates over comments, sends each comment’s text to a GPT model via the OpenAI API together with a prompt, and records the model’s constrained outputs for analysis and audit. The excerpts below reproduce the prompts used in that pipeline (with each comment’s text replaced by a {TEXT} placeholder).

A. Direction of requested regulatory change (5D scheme)

System prompt

You analyze Brazilian regulatory consultation comments and classify what the comment is asking for relative to the proposed regulation (i.e., whether they ask for 'more' or 'less' rigorous regulation). Return only the five integer codes (-1, 0, 1) according to the definitions below. Be conservative—if the comment is unclear or off-topic for a dimension, return 0.

Definitions (always return -1, 0, or 1):

- 1) entities: Does the comment request that the number of regulated entities (e.g., facilities, companies, industries) increases (1), decreases (-1), or stays the same (0)?
- 2) outcomes: Does the comment request that the number of outcomes (e.g., activities, substances) being regulated increases (1), decreases (-1), or stays the same (0)?
- 3) levels: Does the comment request that the level of outcomes (e.g., quality standards) being regulated increases (1), decreases (-1), or stays the same (0)?
- 4) compliance_deadline: Does the comment request that compliance/effective deadlines move to an earlier date (1), a later date (-1), or stay the same (0)?
- 5) monitoring_reporting: Does the comment request stricter (1), more lenient (-1), or the same (0) monitoring and reporting requirements?

Return only integers (-1, 0, or 1) for each field.

TEXT:

{TEXT}

B. Argument-type indicators

System prompt

You analyze Brazilian regulatory consultation comments and classify whether the comment meaningfully employs the following TYPES OF ARGUMENTS. Return ONLY binary flags (1 or 0) using the definitions:

- 1) economic_argument (1/0): Mark 1 if the comment substantively uses economic reasoning, e.g., costs, benefits, efficiency, compliance costs/burdens, investment incentives, price effects, competition, productivity, feasibility/cost-benefit tradeoffs, market impacts. Incidental words like “cost” without an argument do not count.
- 2) legal_argument (1/0): Mark 1 if the comment substantively uses legal reasoning, e.g., cites or relies on laws, decrees, agency resolutions, jurisprudence/case law, constitutional/administrative competence, due process, legal certainty, legality/ultra vires challenges, or other legal grounds.

3) technical_argument (1/0): Mark 1 if the comment substantively uses technical/engineering or telecom network reasoning, e.g., references to spectrum, RF, modulation, bandwidth, latency, interference, QoS/QoE, standards (e.g., ITU/3GPP/ABNT), network architecture, interoperability, equipment specs, safety/performance metrics.

Guidance: Mark 1 only when the reasoning is a meaningful part of the comment; be conservative. If unclear, mark 0. Return exactly the three integers (1 or 0), one for each field.

TEXT:

{TEXT}