AI-Powered Qualitative Coding for Evaluating School Restorative Justice in Bogotá

Felipe Nuñez
Departamento de Ingeniería de Sistemas
Universidad de los Andes
Bogotá, Colombia
f.nunez@uniandes.edu.co

Abstract

School Restorative Justice (*Justicia Escolar Restaurativa*, JER) is a district-wide strategy in Bogotá that seeks to move from punitive discipline towards dialogue, repair of harm, and peace-building in schools. As part of its evaluation, the Secretaría de Educación del Distrito (SED) collected 91 semi-structured interviews across 13 public schools and three district officials, involving principals, teachers, students, families, and SED staff. Traditional qualitative coding of this volume of data would require several months of work and multiple coders to achieve reliability.

This paper presents the design and implementation of an AI-powered system that automates the first pass of qualitative coding for this corpus. The system combines a 55-code analytical framework (grouped into 13 thematic categories) with a multi-stage natural language processing (NLP) pipeline based on GPT-4 and text embeddings. The pipeline performs interview cleaning, segmentation into "coding units", vectorization, two-stage code selection, and threshold-based quality control, producing structured coded data and a complete audit trail.

On the JER corpus, the system processes the 91 interviews in approximately 48 hours on first run (and 24 hours with cached embeddings), reaching a classification rate of 7090% of fragments with at least one high-confidence code and mean confidence scores between 0.76 and 0.88. The results demonstrate that large language models can be integrated into a rigorously designed, transparent pipeline that accelerates qualitative analysis while preserving conceptual depth and methodological caution in sensitive public policy contexts.

CCS Concepts

• **Natural language processing**; • **Information systems** → *Data mining*;

Keywords

Natural language processing, qualitative analysis, large language models, education policy, restorative justice

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Felipe Nuñez. 2025. AI-Powered Qualitative Coding for Evaluating School Restorative Justice in Bogotá. In . ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

School Restorative Justice (*Justicia Escolar Restaurativa*, JER) is a policy initiative in Bogotá that promotes non-punitive approaches to school conflict, emphasizing dialogue, recognition of harm, and the reconstruction of trust within school communities. The programme seeks to strengthen socioemotional and civic capacities in students, transform school culture around coexistence, and consolidate schools as territories of peace.

To evaluate how JER is implemented and perceived in practice, the Secretaría de Educación del Distrito (SED) carried out 91 semi-structured interviews across 13 public schools and three district officials. The sample covered six stakeholder groups: principals, teachers and counsellors, elementary students, middle/high school students, families, and SED officials. The interviews explore perceptions of JER, concrete restorative practices, changes in coexistence, institutional transformations, and recommendations for sustaining the initiative.

From a methodological standpoint, this corpus is ideal for qualitative analysis: interviews are rich, multi-perspective, and directly connected to the programme's theory of change. However, traditional manual coding with tools such as Atlas.ti or NVivo would require 36 months of work, multiple coders for reliability, and substantial coordination and documentation effort. In addition, public sector evaluations often operate under tight time and budget constraints, which makes it difficult to combine thoroughness and timeliness.

Recent advances in large language models (LLMs) such as GPT-4 open new possibilities for supporting qualitative research. Given clear instructions, exemplars, and a high-quality codebook, these models can perform classification-like tasks that resemble thematic coding. At the same time, their use in education and public policy requires special care: issues of transparency, bias, reproducibility, and alignment with the original analytical framework become critical.

In this work, we present an AI-powered qualitative coding system developed specifically for the JER evaluation. The system is designed as an engineering artefact that bridges rigorous qualitative methodology and modern NLP. It takes as input raw interview transcripts, and outputs cleaned texts, segmented fragments (coding units), and JSON files where each fragment is associated with 03 thematic codes, confidence scores, and detailed logs of model reasoning and validation steps.

The contributions of this paper are threefold:

- We formalize the JER evaluation framework into a 55-code analytical structure grouped in 13 thematic categories, suitable for machine-assisted coding.
- We design and implement an end-to-end NLP pipeline that combines GPT-4, text embeddings, and explicit thresholds to produce conservative, auditable first-pass coding for 91 interviews.
- We report empirical results on processing time, coverage, confidence levels, and generated outputs, and discuss methodological implications, limitations, and avenues for humanAI collaboration in educational policy evaluation.

The remainder of the paper is organized as follows. Section 2 summarizes the evaluation context and analytical framework. Section 3 describes the dataset. Section 4 details the system architecture and pipeline stages. Section 5 presents results and outputs. Section 6 discusses implications and limitations. Section 7 concludes and outlines future work.

2 Background and Analytical Framework

2.1 School Restorative Justice in Bogotá

The JER programme in Bogotá aims to shift school coexistence from punitive disciplinary logics towards restorative approaches. Its core principles include: (i) dialogue-based conflict resolution; (ii) recognition and reparation of harm; (iii) participation of those affected; (iv) peace as a fundamental right; and (v) the integration of truth and memory related to Colombia's armed conflict into school life.

At the institutional level, JER is expected to generate changes in school norms and practices (e.g., coexistence manuals, institutional education projects, budget allocation, teacher training), as well as in the lived experience of students, families, and staff. Evaluating whether and how these changes occur requires connecting everyday narratives from school actors with the programme's theory of change.

2.2 Qualitative Evaluation and the 55-Code Framework

To operationalize the evaluation, the team developed a structured codebook with 55 thematic codes, grouped into 13 major categories. Each code is defined by: (i) a formal definition aligned with the evaluation framework; (ii) a set of atomic keywords; (iii) strong semantic synonyms; (iv) example phrases with low abstraction level; and (v) high-level examples that capture the intended meaning even without keywords.

The 13 categories are:

(1) **Context** (7 codes): school context, prior programmes, COVID-19 impacts, baseline coexistence conditions.
(2) **Peace as Right** (5 codes): conceptualizations of peace, its defence, relation with other rights.
(3) **Restorative Approach** (6 codes): recognition of harm, attitudes towards JER, restorative practices, reconciliation.
(4) **Socioemotional Capacities** (2 codes): changes and actions that strengthen emotional self-regulation, empathy, and care.
(5) **Civic Capacities** (2 codes): participation, citizenship, democratic practices and decision-making.
(6) **Truth and Memory** (2 codes): references to the armed conflict, historical memory, and curriculum integration.
(7) **Pedagogical Integration** (1 code): incorporation of JER in teaching practices and classroom work.
(8) **Trust Reconstruction** (1 code): rebuilding damaged relationships and confidence among actors.
(9) **School Environment** (4 codes): perceptions of security, changes in climate, schools as territories of peace.
(10) **Coexistence** (3 codes): experiences, actions and significant changes in everyday coexistence.
(11) **Restorative Processes** (4 codes): processes of harm recognition, dialogue, agreements and follow-up.
(12) **Institutional Changes** (12 codes): transformations in institutional projects, manuals, budget, curriculum and training.
(13) **Recommendations** (4 codes): sustainability, improvement proposals and challenges for JER.

Code definitions are intentionally richer than a simple keyword list. This design encourages models to focus on whether a fragment *develops* the concept (for example, describing a concrete restorative circle or a change in school climate), rather than merely mentioning related vocabulary. The "Restorative practices and conflict resolution" code is treated with particular care, as it tends to be over-assigned if the distinction between generic talk about conflict and concrete restorative practices is not enforced.

Table 1: Interview participants by group

Group	Role (Spanish)	Count	Purpose
Principals	Directivos	13	Institutional perspective
Teachers	Docentes y Orientadores	13	Implementation perspective
Elementary students	Estudiantes Primaria	26	Student experience (younger)
Middle/high students	Estudiantes Bachillerato	26	Student experience (older)
Families	Familias	10	Family perspective
SED officials	Funcionarios SED	3	Policy and programme view
Total		**91**	

3 Data and Case Study

3.1 Interview Corpus

The dataset consists of 91 semi-structured interviews collected between March and May 2025, distributed across 13 public educational institutions in Bogotá and three officials from SED. Table 1 summarizes the participant groups.

Interviews follow a questionanswer format, with a common core of questions adapted to each stakeholder type. Topics include

knowledge and perceptions of JER, concrete experiences of restorative practices, changes in coexistence, institutional adjustments, and perceived challenges and recommendations.

3.2 Ethical and Operational Considerations

Interviews were conducted under the ethical and confidentiality guidelines of SED. Personal identifiers and highly sensitive details are not stored in the analytic dataset. In some cases, the research team had limited access to auxiliary segmentation variables (for example, exact school or group IDs) to avoid any potential re-identification. This constraint reinforced the need for a methodology that relies primarily on textual content for analysis.

The corpus is heterogeneous: students, families and staff express themselves with different levels of abstraction and vocabulary, and there is variability in transcription quality. These characteristics informed the design of the cleaning and segmentation stages of the pipeline.

4 Methodology: Automated Qualitative Coding System

4.1 Design Principles

The system was designed under five guiding principles:

- **Conceptual fidelity**: the pipeline must reflect the original evaluation framework and codebook, rather than inventing new categories.
- **Transparency**: each decision about codes should be traceable via logs, scores and explicit rules.
- **Conservativeness**: codes should only be assigned when there is strong textual evidence; unclear cases are left uncoded for human review.
- **Modularity**: stages (cleaning, segmentation, classification, analysis) must be separable and reusable.
- **Practical efficiency**: the system must process the full corpus within hours and be operable on standard hardware with access to the OpenAI API.

The implementation is written in Python 3.8+ and uses the GPT-4 model for text understanding and classification, the `text-embedding-ada-002` model for vectorization, and NumPy for numerical operations.

4.2 Pipeline Overview

As summarized in Figure 1, the pipeline consists of six main stages:

- **Stage 1 Data ingestion**: read raw `.txt` transcripts in questionanswer format.
- **Stage 2 Cleaning**: remove non-analytic noise while preserving all participant content.
- **Stage 3 Segmentation**: split long answers into 03 semantically coherent fragments (coding units).
- **Stage 4 Vectorization**: compute and cache 1 536-dimensional embeddings for each fragment.
- **Stage 5 Two-stage classification**: (5A) fast candidate filtering and (5B) deep expert analysis with GPT-4.
- **Stage 6 Quality control and export**: apply thresholds and rules, store coded fragments and logs as JSON.

High-level schematic of the automated qualitative coding pipeline for the JER interviews.

Figure 1: Conceptual overview of the automated qualitative coding system.

For space reasons we describe the main stages at a high level; full prompts, parameters and file formats are documented in the project files.

4.3 Stage 2: Interview Cleaning

The cleaning stage uses GPT-4 with a conservative prompt and temperature 0.2. The model receives the raw transcript and is instructed to:

- remove page numbers, headers, footers, image captions and technical metadata;
- preserve all participant answers, including repetitions that carry meaning;
- maintain the original questionanswer structure.

The output of this step is a cleaned `.txt` file per interview. No coding or summarization are performed at this point; the goal is only to reduce noise without semantic loss.

4.4 Stage 3: Segmentation into Coding Units

Segmentation transforms cleaned interviews into a list of fragments that are appropriate for thematic coding. It is also implemented with GPT-4 (temperature 0.2) under explicit rules:

- Texts shorter than 500 characters are not segmented.
- Each answer can be split into at most three fragments.
- Each fragment must have at least 150 characters.
- Splits are only allowed when the answer clearly develops different concepts that could correspond to different codes.

The output is a JSON file per interview where each item stores the original question, the fragment text, and metadata (e.g., position within the interview). The design emulates the notion of "quotations" in Atlas.ti, but delegates the tedious splitting to the model under tight constraints.

4.5 Stage 4: Vectorization and Embedding Cache

Each fragment is converted into a fixed-length embedding using the `text-embedding-ada-002` model (1 536 dimensions). To avoid redundant API calls, embeddings are stored in a persistent cache (`embeddings_cache.pkl`) keyed by fragment text.

On a full run, the cache size ranges between roughly 50100 MB for the corpus, and the complete set of pipeline outputs (cleaned texts, segments, classification results, logs and embeddings) requires on the order of 220445 MB of storage, depending on configuration.

4.6 Stage 5: Two-Stage Classification with GPT-4

The core of the system is a two-stage expert analysis procedure.

Stage 5A: Rapid filtering. In the first stage, GPT-4 receives a fragment and the full list of 55 codes with their definitions. With a short, focused prompt and low temperature (0.1), the model is asked to identify a small subset (typically 58) of codes that could

plausibly apply to the fragment. The goal is to quickly exclude clearly irrelevant codes, not to make final decisions.

Stage 5B: Expert analysis. The second stage takes the fragment and the candidate codes from Stage 5A and runs a deeper analysis with GPT-4 (temperature 0.07, higher token limit). For each candidate, the model is instructed to consider whether the fragment *develops* the concept in the code definition (instead of merely mentioning related words), to provide a natural-language justification, and to assign a confidence score between 0 and 1. The model can assign between 0 and 3 final codes to each fragment.

The system computes an overall quality score for each classification that combines internal signals (e.g., clarity of justification, consistency between codes) into a scalar in $[0, 1]$.

4.7 Quality Control and Fallback Mechanism

To control precision, three main thresholds are used:

- `SIMILARITY_THRESHOLD` = 0.65 for cosine similarity between fragment embeddings and code descriptions (used in auxiliary candidate selection and analysis tools).
- `API_CONFIDENCE_THRESHOLD` = 0.76 as the minimum per-code confidence.
- `QUALITY_SCORE_THRESHOLD` = 0.63 as the minimum overall quality score.

A code is accepted for a fragment only if both its confidence and the quality score exceed their respective thresholds; at most three codes are retained per fragment. These default values can be adjusted (more strict or more lenient) depending on validation results.

If the GPT-4-based stages fail for technical reasons, a fallback mechanism uses the precomputed embeddings. For each fragment, the system computes cosine similarity between the fragment and all code embeddings, optionally combined with Euclidean distance, and selects up to two codes that exceed a high similarity threshold (e.g., 0.84). The confidences of fallback codes are calibrated downwards to reflect their secondary nature. In practice, this mechanism is used sparingly as a safety net.

4.8 Outputs

The pipeline produces four main groups of artefacts:

(1) **Cleaned transcripts**: 91 text files with minimal noise, one per interview.
(2) **Segmented fragments**: JSON files (66 in the current configuration) containing lists of coding units per interview.
(3) **Classified fragments**: JSON files (64 in the current configuration) where each fragment is associated with a set of finalized codes, their confidence scores, the originating question, and the full reasoning trace.
(4) **Logs and caches**: embeddings cache, intermediate decisions, and summary statistics for monitoring.

These outputs support downstream analysis by stakeholder group, school, and thematic category using standard data science tools.

Table 2: System performance metrics on the JER corpus

Metric	Value
Processing time (first run)	48 hours
Processing time (with cache)	24 hours
Storage required (all outputs)	~220445 MB
Approximate number of API calls	56 000112 000
Classification rate[1]	7090%
Average confidence (accepted codes)	0.760.88

5 Results

5.1 Processing Performance

Table 2 summarizes key performance metrics for the current configuration.

Compared with a manual baseline that would require several months of full-time work, the pipeline processes the entire corpus in less than a working day on standard hardware, including all cleaning, segmentation, classification and logging.

5.2 Coverage and Coding Behaviour

The classification rate between 70% and 90% indicates that the system is able to assign at least one high-confidence code to the majority of fragments, while leaving a non-trivial proportion uncoded for cautious handling. This is consistent with the conservative design: uncertain cases are flagged rather than forced into categories.

An inspection of model outputs shows that:

- Fragments that explicitly describe restorative circles, mediated dialogues or agreements are reliably coded under "Restorative processes" and related subcodes in the "Restorative Approach" category.
- Narratives about changes in school climate, improved relationships and schools as peace territories are captured by codes in the "School Environment" and "Trust Reconstruction" categories.
- Some borderline cases arise when fragments speak about conflict or sanctions in general terms without clearly describing restorative practices. In those cases, the thresholds and explicit instructions for the "restorative practices" code reduce over-assignment.

Because the full reasoning and justifications are stored, analysts can audit model decisions and refine thresholds or prompts over time.

5.3 Analytical Possibilities

The coded dataset enables several analytical cuts that are usually difficult to obtain at scale:

- **By stakeholder group**: comparison of how principals, teachers, students, families and SED officials describe JER, which themes they highlight, and where perceptions diverge.
- **By school**: identification of schools where institutional changes (e.g., in coexistence manuals or teacher training) are more

[1]Share of fragments that receive at least one code above confidence and quality thresholds.

frequently mentioned, or where restorative practices appear more embedded in everyday routines.
- **By thematic category**: assessment of which dimensions of the programme (e.g., socioemotional capacities, truth and memory, pedagogical integration) appear more or less developed in the narratives.

Although the present paper focuses on system design and performance, these analytical outputs provide a basis for more substantive evaluation work with policymakers and school communities.

6 Discussion

6.1 Methodological Implications

From a qualitative methods perspective, the system illustrates how a detailed codebook and a carefully constrained LLM pipeline can provide a realistic compromise between full manual coding and opaque black-box analysis. Instead of replacing human judgement, the pipeline produces a structured, auditable first pass that can guide subsequent human interpretation.

The use of confidence and quality thresholds is central. By enforcing minimum values (e.g., 0.76 for per-code confidence and 0.63 for overall quality), the system avoids presenting weak assignments as definitive findings. Analysts can relax or tighten these thresholds based on validation against human-coded samples, balancing coverage and precision according to the evaluation's needs.

6.2 Technical and Practical Considerations

The technical design is deliberately simple from an engineering standpoint: Python scripts orchestrate calls to the OpenAI API and store results as plain text and JSON, without complex web services or user interfaces. This makes the system easier to maintain, adapt and extend by other research teams.

However, several practical issues remain:

- **Cost and dependencies**: the pipeline depends on commercial APIs and requires a non-trivial one-time processing cost (roughly $500–$1 500 USD depending on exact configuration and pricing). This may be a barrier for some public sector projects.
- **Model opacity**: despite having logs of decisions, the internal representation of GPT-4 remains a black box. Certain systematic biases or failure modes may not be immediately apparent.
- **Context specificity**: prompts, thresholds and the codebook have been tuned for Spanish-language interviews in the Colombian educational context. Direct transfer to other contexts would require adaptation.

6.3 Limitations

The main limitations of the current work are:

- **Limited external validation**: while the system includes internal quality scores and spot checks with domain experts, a full-scale comparison against human-coded gold standards (e.g., inter-coder agreement, precision/recall per code) remains future work.
- **No direct modelling of interactions**: the pipeline treats fragments independently, without using the full conversational structure (e.g., follow-up questions, interactions between actors).
- **Restricted use of demographic or contextual variables**: due to confidentiality constraints, some potentially relevant segmentation variables (e.g., specific school, grade level) cannot be used directly in modelling.

These limitations do not invalidate the utility of the tool as a first-pass coding engine, but they underscore the need for cautious interpretation and complementary qualitative work.

6.4 Opportunities for HumanAI Collaboration

The system opens several avenues for humanAI collaboration in policy evaluation:

- Analysts can use coded outputs to identify segments of interest (e.g., all fragments about "truth and memory" from students) and then perform in-depth, human-led interpretation.
- By inspecting disagreements between model codes and human expectations, teams can iteratively refine prompts, thresholds and even the codebook itself.
- In future iterations, active learning strategies could prioritize human review on low-confidence fragments to improve the system over time.

In this sense, the contribution is less about automating interpretation and more about creating a robust infrastructure that makes qualitative analysis more scalable and systematic.

7 Conclusions and Future Work

This paper has presented an AI-powered qualitative coding system developed for the evaluation of the School Restorative Justice (JER) programme in Bogotá. By formalizing a 55-code analytical framework and integrating GPT-4 and text embeddings into a modular pipeline with explicit thresholds and logging, the system can process 91 semi-structured interviews in a matter of hours while preserving conceptual depth and methodological caution.

The results show that a large proportion of fragments can be assigned high-confidence codes, enabling analyses by stakeholder group, school and thematic dimension that would be difficult to obtain manually under the same time and budget constraints. At the same time, limitations related to model opacity, context specificity and external validation highlight the importance of using such systems as complements-not replacements-to human qualitative expertise.

Future work includes: (i) conducting a systematic evaluation of coding quality against human-coded samples; (ii) experimenting with open-source language models and on-premise deployment to reduce dependency on external APIs; (iii) developing interactive interfaces that allow analysts and policymakers to explore the coded dataset more intuitively; and (iv) adapting the methodology to other educational and social policy evaluations that rely on large volumes of interview or focus group data.

More broadly, the experience in this project suggests that engineering-style design-with clear requirements, modular architecture, and explicit quality criteria-can play a key role in making LLM-based

qualitative analysis both useful and responsible in real-world public sector contexts.

Acknowledgments

The author(s) thank the Secretaría de Educación del Distrito (SED) and the participating schools, students, families and staff for their collaboration and trust. This work was developed in the context of an academic project on AI-assisted qualitative analysis.

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *arXiv preprint* arXiv:2005.14165. https://doi.org/10.48550/arXiv.2005.14165.

[2] Massimiliano Patacchiola, Mingfei Sun, Katja Hofmann, and Richard E. Turner. 2023. Comparing the Efficacy of Fine-Tuning and Meta-Learning for Few-Shot Policy Imitation. *arXiv preprint* arXiv:2306.13554. https://doi.org/10.48550/arXiv.2306.13554.

[3] Yao Ge, Sudeshna Das, Yuting Guo, and Abeed Sarker. 2025. Retrieval Augmented Generation Based Dynamic Prompting for Few-Shot Biomedical Named Entity Recognition Using Large Language Models. *arXiv preprint* arXiv:2508.06504. https://doi.org/10.48550/arXiv.2508.06504.