From Quotes to Concepts: Axial Coding of Political Debates with Ensemble LMs

Angelina Parfenova[1,2], David Graus[3], and Juergen Pfeffer[1]

[1] Technical University of Munich
angelina.parfenova@tum.de
[2] Lucerne University of Applied Sciences and Arts
[3] University of Amsterdam
d.p.graus@uva.nl

Abstract. Axial coding is a commonly used qualitative analysis method that enhances document understanding by organizing sentence-level open codes into broader categories. In this paper, we operationalize *axial* coding with large language models (LLMs). Extending an ensemble-based open coding approach with an LLM moderator, we add an axial coding step that groups open codes into higher-order categories, transforming raw debate transcripts into concise, hierarchical representations. We compare two strategies: (i) clustering embeddings of code-utterance pairs using density-based and partitioning algorithms followed by LLM labeling, and (ii) direct LLM-based grouping of codes and utterances into categories. We apply our method to Dutch parliamentary debates, converting lengthy transcripts into compact, hierarchically structured codes and categories. We evaluate our method using *extrinsic* metrics aligned with human-assigned topic labels (ROUGE-L, cosine, BERTScore), and *intrinsic* metrics describing code groups (coverage, brevity, coherence, novelty, JSD divergence). Our results reveal a trade-off: density-based clustering achieves high coverage and strong cluster alignment, while direct LLM grouping results in higher fine-grained alignment, but lower coverage (∼20%). Overall, clustering maximizes coverage and structural separation, whereas LLM grouping produces more concise, interpretable, and semantically aligned categories. To support future research, we publicly release the full dataset of utterances and codes, enabling reproducibility and comparative studies.

Keywords: Qualitative Coding · Axial Coding · Political Debates · Language Models

1 Introduction

Automated methods for sense-making of large text corpora have a long tradition in information retrieval, in particular in domains where human review or exploratory search is central, such as legal search [6], e-discovery [19], and patent retrieval [15]. In the social sciences, *qualitative coding* has established itself as

the de facto method for making large text collections interpretable. Here, human coders read text, assign short labels that capture its meaning (*open coding*), and then compare, refine, and group these labels, typically across documents (*axial coding*). While coding proves flexible in analyzing lengthy documents such as interview transcripts, political debates, or survey responses, it is known to be slow, subjective, and difficult to scale [14].

Recently, large language models (LLMs) have been shown to support qualitative coding by successfully generating plausible *open codes* – short descriptive labels for individual text segments [9,21,28]. These approaches, however, omit the *axial coding* step of grouping codes into higher-order categories, an abstraction stage that is central in qualitative analysis but has not been operationalized in prior IR or NLP methods.

This paper introduces the first method that scales axial coding using LLMs, enabling the automated qualitative analysis of large-scale textual corpora. Our method extends an existing ensemble-based open coding framework [22] with an axial coding method to produce higher-order labeled code categories. Hence, our method consists of two steps: (i) *open coding*: generate concise open codes using a moderator-style ensemble with label refinement, and (ii) *axial coding*: group codes into higher-order labeled categories using one of two strategies: unsupervised clustering of embeddings (described in §4.1), and LLM-based grouping (in §4.2). To illustrate the output of our method, we show a compact concept map in Fig. 2, which visualizes how raw utterances (grey) are assigned open codes (green) and then grouped into axial categories (yellow).

We validate our method using transcripts of Dutch parliamentary debates, a challenging domain with heterogeneous speakers, procedural dialogue, evolving topics, and lengthy documents. To assess both the fidelity to human annotations and interpretability of the resulting groups, we evaluate our method using two approaches: *extrinsic* evaluation, where we measure alignment with human-assigned labels using similarity metrics such as ROUGE [**?**], cosine similarity, and BERTScore [30]; and *intrinsic* evaluation, where we use metrics independent of gold labels that capture properties deemed important for qualitative analysis, including coverage, label brevity, coherence, novelty, and divergence [5]. This dual evaluation allows us to assess generated categories even where gold standards are partial or contested, highlighting how axial coding with LLMs can scale qualitative analysis while retaining interpretability.

Our contributions are three-fold: (1) We introduce the first approach to scale *axial coding* with LLMs, extending ensemble-based open coding with two alternative strategies: (i) clustering code embeddings with LLM-based labeling, and (ii) direct grouping of codes by LLMs. (2) We design a dual evaluation scheme that combines *extrinsic* alignment with human-coded taxonomies (ROUGE, cosine, BERTScore) and *intrinsic* interpretability metrics (coverage, brevity, coherence, novelty, divergence), enabling systematic comparison of clustering- vs. LLM-based axial coding. (3) We publicly release a dataset of 5,000 debate utterances from Dutch parliamentary transcripts, including human-assigned domain/subtopic labels, LLM-generated open codes, and axial categories from both

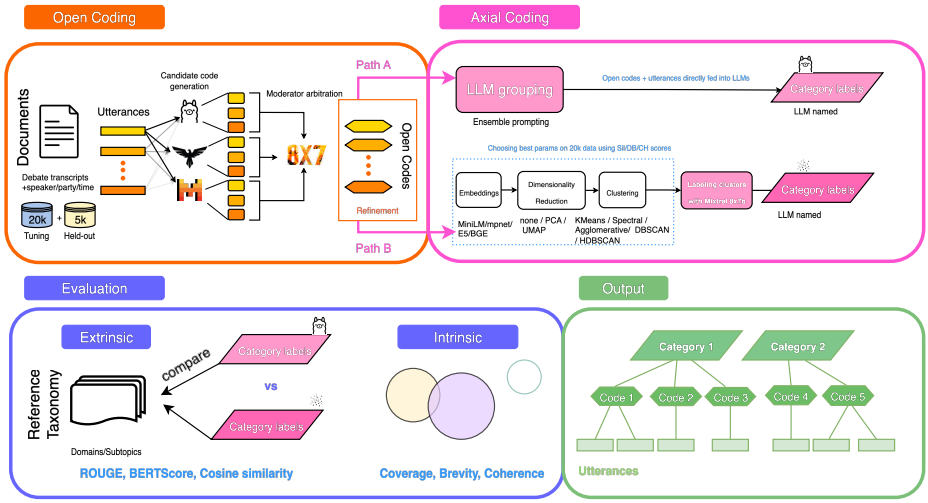

Fig. 1. Pipeline for open and axial coding. In the open coding stage (orange box), we produce open codes for utterances from transcripts (20k train set for finding best parameters, 5k held-out set), using an ensemble of LoRA-finetuned LLMs with a moderator model, following the framework of [22]. The axial coding stage (pink box) groups them into categories using one of two methods: direct grouping with LLM prompting (path A), or clustering embeddings, followed by LLM labeling (path B). Evaluation (blue box) covers both *extrinsic* alignment with human-assigned domain labels and *intrinsic* interpretability metrics (coverage, brevity, coherence, novelty, divergence). The output (green box) is a structured mapping from utterances to codes and categories, see also Fig. 2.

clustering and LLM methods, to support reproducibility and comparative studies.[4]

2 Background

Qualitative data analysis (QDA) is a central method in the social sciences to identify and interpret patterns in unstructured text [18,7]. A core step is *coding*: assigning short essence-capturing labels to text segments [26]. These codes form the building blocks for higher-level categories and themes [4]. Coding is typically an iterative and collaborative process, involving negotiation and refinement by teams of analysts. In QDA, two main stages of coding are commonly distinguished: (i) *open coding* generates initial codes that capture the essence of text segments, and (ii) *axial coding* organizes these codes into higher-order categories [26]. Axial coding originates from grounded theory methodology and has been widely adopted in qualitative research for theory construction, category refinement, and comparative analysis across cases [10]. Coding can proceed

[4] `https://github.com/Likich/axial_coding_dataset`

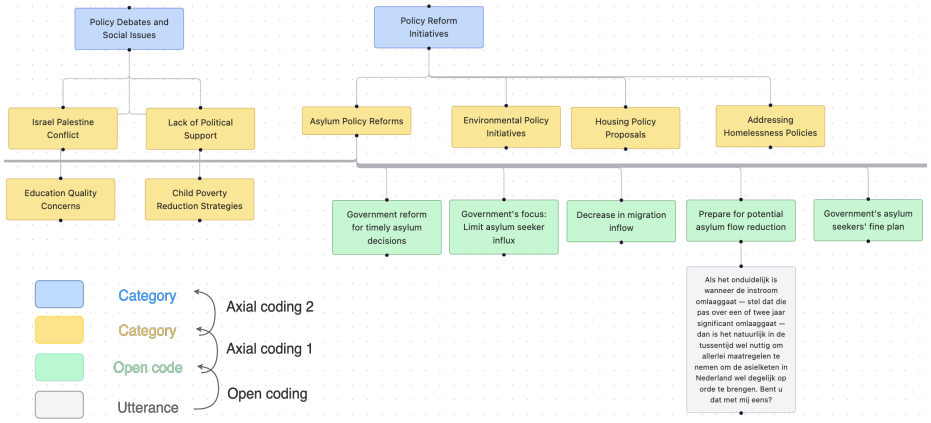

Fig. 2. Example concept map based on the subset of results with HDBSCAN on the 5k held-out set. Utterances (grey) → open codes (green) → first-order axial categories (yellow) → second-order aggregates (blue). The second-order aggregates were obtained by applying the same grouping on the first-order categories; this is illustrative (not used in the evaluation), but helps convey the hierarchical nature of the axial coding process.

inductively, allowing codes to emerge organically from the material (in line with grounded theory traditions [10]), or *deductively*, by applying a predefined set of labels. Qualitative coding enables researchers to turn raw text into structured and interpretable data, and presents a complementary perspective to quantitative analysis methods commonly used in information retrieval [8].

LLMs for qualitative coding. Recent work has explored large language models (LLMs) as tools to automate or support coding [3,13,21]. LLMs can produce plausible open codes and accelerate inductive or deductive coding [9,17,27,29,31]. However, these methods have been limited to open coding, and no prior work has operationalized axial coding with LLMs to provide the abstraction step that groups codes to further facilitate analysis and human review.

Quantitative analysis of political discourse. The IR community has a long history in studying quantitative methods for analyzing and structuring large, heterogeneous text collections such as parliamentary debates. Foundational methods such as topic modeling [2] and embedding-clustering pipelines (e.g., BERTopic [11]) offer unsupervised ways to identify topics and structures in large text corpora.

While axial coding may appear superficially similar to coarse-grained topic modeling, the two differ in both intent and output. Topic models induce latent word distributions optimized for corpus-level structure [2,20], whereas axial coding operates over *interpreted semantic units* (codes) and aims to construct human-readable, reusable conceptual categories. Axial categories are not latent variables but named abstractions intended for analytic reuse, comparison, and

theory building. From this perspective, axial coding can be seen as an interpretive layer that operates on top of utterance-level codes rather than directly on token distributions, and thus complements topic modeling rather than replicating it.

Parliamentary debate graph extraction [25], modeling *"who said what to whom"* [12], and visualization systems such as *ThemeStreams* [24] and *ATR-Vis* [16] demonstrate how IR methods can facilitate exploration and sense-making of political debates. Topic modeling-based methods are effective in identifying broad themes, but quality issues such as *topic impurity* and *topic generality* can make them difficult to interpret or reuse across studies [1]. In addition, by focusing on coarse latent themes, quantitative methods risk losing the depth and contextual richness that qualitative methods can provide [8]. Our method induces fine-grained, interpretable codes at the utterance level, and organizes them into hierarchical categories, yielding structures that are more directly comparable to human-coded taxonomies, and potentially more useful for reuse and interpretive analysis.

Positioning. Our work builds on both the traditions of QDA in social sciences and the quantitative methods from IR: from QDA we take the notion of *inductive* and *axial coding*; from IR we inherit clustering methods, evaluation metrics, and graph-based modeling of debates. By combining ensemble LLM-based open coding with clustering- and LLM-driven axial coding, and by representing the results as a *concept graph*, we propose a method that transforms raw debate transcripts into interpretable conceptual structures.

3 Dataset and Open Coding Method

Our method extends an existing method for open coding using ensemble LLMs [22], which we describe in §3.1, below. We use a dataset created from Dutch parliamentary debate transcripts, with utterance-level metadata (speaker, party, role), document metadata (meeting title, date, `source_doc_id`), and associated topic labels from a human-curated taxonomy from the Dutch national government, consisting of 17 domains and 79 subtopics which are assigned at the document level.[5] We use these labels as ground truth for evaluation, rather than as direct supervision for coding. Table 1 summarizes key statistics of the datasets we use.

This corpus is particularly suitable for studying axial coding because debates are lengthy, multi-topic transcripts where abstraction into higher-order categories is naturally required for human review. In addition, the availability of a human-curated domain/subtopic taxonomy enables extrinsic evaluation without treating the taxonomy as supervision.

3.1 Open Coding

The first stage of our pipeline produces *open codes* for each utterance: short labels that capture their essence. This first stage is illustrated in the orange box

[5] `https://www.rijksoverheid.nl/onderwerpen/`

Table 1. Dataset statistics for the full corpus of Dutch parliamentary debates, the 20k training subset, and the 5k held-out test set. Number of utterances is reported per debate. Utterance length values are reported as mean / max words. Document length refers to the sum of all utterance lengths in a single debate. Speaker values are reported both globally and per debate. Topic counts are reported both at the domain level and the fine-grained subtopic level.

Split	Uttrs	Spkrs	Part.	Docs	Spkrs per doc (mean/max)	Utt. per doc	Utt. len (mean/max)	Doc. len (mean/max)	Top./ Subtop.
Full	82,732	248	17	2,038	4.7/77	40.6	119/7,060	4,844/124,352	17/79
Train	20,000	211	17	578	4.0/59	34.6	138/7,060	4,760/83,863	17/53
Test	5,000	128	15	196	3.8/57	25.5	125/2,022	3,178/50,129	16/42

of Fig. 1. We implement the ensemble-based coding framework of Parfenova and Pfeffer [22], which mirrors how human coders typically work: multiple annotators propose candidate codes which are refined through team discussion to improve consistency. In our setup, multiple smaller specialized LLMs generate candidate codes, a moderator LLM arbitrates between them, and a post-hoc refinement step consolidates the outputs. This approach was shown to improve both stability and interpretability compared to using individual models alone.

Candidate code generation. We use instruction-tuned LLMs (e.g., LLaMA-3, Falcon, Mistral), adapted to the open coding task using Low-Rank Adaptation (LoRA), with a dataset of 1,000 *quote-code* pairs from social science research studies and the SemEval-2014 Task 4 dataset [23]. Each model independently proposes a concise candidate label (1–5 words) for a given utterance, following prompts that encourage semantic abstraction rather than verbatim repetition.

Moderator arbitration. The moderator is an LLM prompted to act as a lead coder: it is shown the original text segment and candidate codes from the fine-tuned open coding LLMs, and is asked to select the most appropriate one (or propose a refined alternative). In this way, the moderator provides a clear decision, and produces more consistent and stable code assignments than simple ensembling.

Label refinement. To reduce code redundancy, we apply an iterative refinement process. Each newly generated code is compared against all previously assigned labels, using cosine similarity between SBERT embeddings. If the new code's similarity to a previous code exceeds a threshold ($\tau = 0.7$), the previous code is reused; otherwise, the new code is introduced. This threshold follows prior work on semantic similarity in qualitative coding, where 0.7 was empirically found to balance over-merging and under-merging labels [22]. This mechanism promotes label reuse, resulting in shorter average code lengths and more consistent codes.

Outputs. The result of this process is a single refined open code per utterance.

4 Axial Coding Method

The open codes resulting from the coding stage described above serve as input to our novel axial coding stage, illustrated in the pink box in Fig. 1. We compare two methods for grouping codes into higher-order categories: (i) *unsupervised clustering* in embedding space, and (ii) *LLM-powered grouping*, where LLMs directly propose and name categories.

4.1 Clustering-based axial coding

We implement a clustering-based axial coding method, by embedding both the *open code* and its associated *utterance* jointly. This balances interpretability of concise labels with richer contextual information.

For embedding the input, we tested a range of sentence encoders: `all-MiniLM-L6-v2`, `multi-qa-MiniLM-L6-cos-v1`, `paraphrase-MiniLM-L3-v2`, `all-mpnet-base-v2`, and the `e5` and `bge` families (small/base/large). Dimensionality reduction was varied between none, PCA (64 dimensions), and UMAP (15 dimensions). For the actual clustering, we compared algorithms including KMeans, Agglomerative, Spectral, Gaussian Mixture Models (GMM), and density-based methods: DBSCAN and HDBSCAN. For density-based methods, we use *ms* to denote DBSCAN's `min_samples` parameter, and *mcs* to denote HDBSCAN's `min_cluster_size`. For all other methods, we fix the number of clusters to $k \in \{4, 6, \ldots, 20\}$, which roughly aligns with the 17 domain-level categories in the human taxonomy. This mirrors topic modeling workflows while retaining code-level interpretability.

To identify which clustering approach performs best, we first conduct a large-scale sweep on the 20k training set, ranking by intrinsic clustering metrics: silhouette score, Davies–Bouldin Index (DBI), and Calinski–Harabasz Index (CHI). Top-performing configurations are reported in Table 2, and reapplied to the 5k held-out set to assess whether they generalize well to unseen utterances (Table 3).

Results. We find that density-based methods (DBSCAN and HDBSCAN) achieve the highest silhouette scores, reflecting compact, well-separated clusters. However, this comes at the cost of coverage: DBSCAN often discards over 85–95% of utterances as noise. The resulting clusters are small and clean, but unrepresentative of the debate corpus. Other clustering models do not discard utterances as much, providing full coverage, but yield less separated clusters according to the Silhouette score. Overall, we adopt both density-based clustering methods (DBSCAN and HDBSCAN) in our axial coding method, since they balance coverage with flexible discovery of cluster counts, while still achieving strong internal scores. The high noise rates of density-based methods reflect conservative cluster formation rather than failure, mirroring qualitative coding practices where ambiguous material may be left uncategorized during early abstraction stages.

Table 2. Top clustering configurations on the 20k training set. For KMeans, Agglomerative, Spectral, and GMM we report the standard silhouette score. For density-based methods (DBSCAN, HDBSCAN), we report silhouette computed only on core points (excluding noise), since global silhouette is dominated by the noise label.

Embedding	Reduction	Algo	Params	Silhouette	DBI	CHI	Clusters	Noise rate
all-MiniLM-L6-v2	none	DBSCAN	eps=0.3, min_samples=20	0.747	1.057	65.6	18	0.956
all-mpnet-base-v2	none	DBSCAN	eps=0.3, min_samples=20	0.731	1.097	65.8	17	0.955
all-MiniLM-L6-v2	none	DBSCAN	eps=0.3, min_samples=10	0.707	1.066	40.2	41	0.939
paraphrase-MiniLM-L3-v2	UMAP	HDBSCAN	in_cluster_size=10	0.668	0.786	278.0	302	0.504
all-MiniLM-L6-v2	UMAP	HDBSCAN	min_cluster_size=20	0.667	1.026	265.0	151	0.507
bge-base-en	UMAP	HDBSCAN	min_cluster_size=50	0.664	1.086	514.2	55	0.560
e5-small	UMAP	KMeans	k=4, n_clusters=4	0.571	0.615	11982	4	0
e5-small	UMAP	GMM	k=4, n_components=4	0.530	0.687	9213	4	0
all-mpnet-base-v2	UMAP	Spectral	k=4, n_clusters=4, affinity=nn	0.484	1.476	112.6	4	0
e5-small	UMAP	Agglomerative	k=8, n_clusters=8	0.477	0.701	9957	8	0

Table 3. Performance of top clustering configurations on the 5k held-out set. Silhouette is computed only on core points (excluding noise), since global silhouette is dominated by the noise label. Other values include Davies–Bouldin Index (DBI), Calinski–Harabasz Index (CHI), number of clusters, number of noise points, and proportion of points labeled as noise.

Algorithm	Params	Silhouette	DBI	CHI	Clusters	Noise points	Noise rate
DBSCAN	eps=0.3, min_samples=20	0.809	0.979	111.6	6	4,484	0.897
DBSCAN	eps=0.3, min_samples=10	0.657	1.037	63.5	13	4,390	0.878
HDBSCAN	min_cluster_size=20	0.646	1.119	310.9	49	2,394	0.479
DBSCAN	eps=0.3, min_samples=20	0.635	1.079	93.1	7	4,433	0.887
HDBSCAN	min_cluster_size=10	0.612	0.905	488.3	83	1,985	0.397
HDBSCAN	min_cluster_size=50	0.511	1.283	1417.2	3	200	0.040

4.2 LLM-based axial coding

Each utterance–code pair is formatted as a compact item string: `Code: "..." Utterance: "..."`. We process the corpus in medium-sized batches (200–500 items) to balance context window limits with coverage, and pass each batch to an LLM with the instruction to *group items into higher-level categories*. The model returns a JSON object of the form `{"category_label": [code_1, code_2, ...], ...}`, with category labels limited to at most five words.

We run three LLMs independently over the full corpus: Deepseek-R1, Llama-4 Maverick, and Mixtral 8×7B. The outputs across batches are then merged using *label similarity* (cosine similarity between SBERT embeddings of category labels, using *all-MiniLM-L6-v2*) and *membership overlap* (Jaccard similarity between item sets).

Categories are merged if their labels reach a similarity threshold (≥ 0.70 cosine, similarly to open code merging) or a memberships overlap threshold (≥ 0.20 Jaccard). From each merge set, the shortest category label is chosen as the representative name. Each utterance is placed into a single category, defaulting to the largest category in cases of multiple assignments. Importantly, LLMs are not forced to assign each utterance to a category. In practice, some utterances may be left ungrouped when the model does not include them in any category, or when they are not retained during the category merging step. These are marked as *Unassigned* and counted in our **Coverage** metric, reflecting

the expected "abstention" behavior of LLMs. This abstention behavior mirrors human axial coding practice, where not all codes are immediately assigned to higher-order categories, particularly when semantic fit is weak or ambiguous.

5 Evaluation

We assess axial coding performance along two complementary dimensions. Extrinsic evaluation measures alignment with the human-assigned labels of our dataset, providing a reference-based benchmark. Intrinsic evaluation operates without reference labels and relies on unsupervised metrics that capture properties of the extracted categories that are central to qualitative data analysis.

Intrinsic evaluation. First, we evaluate categories independently of a gold taxonomy, focusing on properties that matter for qualitative coding [5]. We use the following measures:

- **Coverage**: the proportion of utterances assigned to categories.
- **Brevity**: the average number of words per category label.
- **Coherence**: the mean textual similarity of utterances within the same category, estimated from MiniLM embeddings.
- **Novelty**: the proportion of singleton categories.
- **Divergence**: the difference of the cluster-size profile (i.e., distributions) of one method with the reference distribution of aggregated cluster sizes.

We measure **divergence** by comparing each method's category-size distribution to the *aggregated code space* (ACS), defined as the mean normalized size distribution across all methods (taken from [5]). Divergence from this distribution is quantified using the Jensen–Shannon distance (JSD), a symmetric measure of distributional difference. A lower JSD indicates that method's categories are more stable and aligned with the ensemble norm.

Together, these metrics operationalize core qualitative concerns: representativeness (coverage), interpretability (brevity), semantic consistency (coherence), abstraction balance (novelty), and structural stability (divergence), enabling evaluation of category systems without relying on a fixed gold taxonomy.

Extrinsic evaluation. Second, we compare our method's outputs to the human-coded taxonomy of debate topics, which consists of two levels: *domains* (16 in our test set) and *subtopics* (42). We compute ROUGE (1/2/L), cosine similarity using MiniLM embeddings, and BERTScore F1, reporting results at both the domain and subtopic level. Importantly, this evaluation relies on (textual) similarity metrics, and not, e.g., classification metrics or accuracy, as the labels produced by our method are inductive codes generated by LLMs, not pre-existing (sub)selected from the taxonomy.

6 Results

Intrinsic evaluation Intrinsic metrics in Table 4 reveal a coverage–coherence trade-off. DBSCAN delivers high intra-cluster coherence (0.93–0.95) but at extremely low coverage (0.10–0.12), making its categories unrepresentative of the corpus. HDBSCAN increases coverage substantially: the BGE+UMAP configuration achieves 96% coverage and the lowest divergence (0.526), but collapses the data into a mere three categories, with moderate coherence (0.661). More balanced settings (e.g., MiniLM+UMAP, Paraphrase+UMAP) discover 49–80 categories with coherence around 0.68–0.69 and similarly moderate divergence, providing finer partitions while avoiding extreme overmerging.

For LLM grouping (bottom three rows), labels are comparatively more concise (1.8–2.9 words on average), and categories align relatively well with consensus size distributions (ACS). Deepseek-R1 shows the lowest divergence (0.134), meaning its partitioning is structurally most similar to the consensus. Llama-4 achieves the highest weighted coherence of the LLM methods (0.576) and the overall lowest novelty (0.024), suggesting cleaner, less fragmented categories, though its coverage (0.328) remains far below the best clustering methods.

Extrinsic evaluation Next, we compare model outputs with the human-labeled domains and subtopics from the taxonomy, see Table 5. Among clustering methods, HDBSCAN with BGE embeddings and UMAP (mcs=50) achieves the strongest alignment on the held-out set.

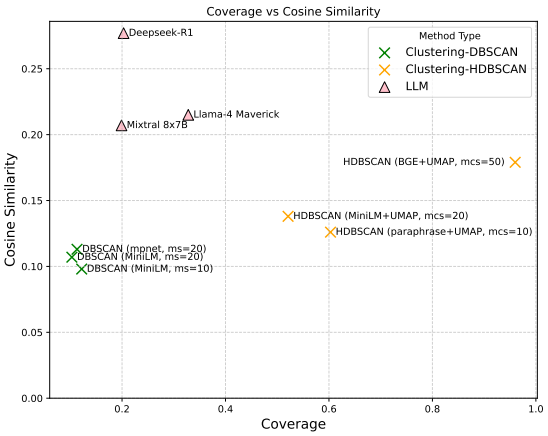

Fig. 3. Coverage vs. cosine similarity for clustering- and LLM-based axial coding methods. Pink triangles denote LLM-based grouping, green crosses denote DBSCAN variants, and orange crosses HDBSCAN variants.

At the subtopic level, it reaches ROUGE-L of 0.111, cosine similarity of 0.300, and BERTScore of 0.867, while covering 96% of utterances (4,800 of 5,000). In contrast, DBSCAN variants achieve near-zero ROUGE because they label around 90% of utterances as noise.

LLM-based grouping improves fine-grained alignment further: Deepseek-R1 attains a ROUGE-L score of 0.248 and cosine similarity of 0.328 at the subtopic level, outperforming the best clustering run on both metrics. However, it assigns categories to only about 20% of utterances, limiting coverage. Llama-4 Maverick

Table 4. Intrinsic evaluation results, showing Coverage (Cov), number of categories (#Cats), Brevity (Brev), Coherence (Coh), Novelty (Nov), and Divergence (Div) for both clustering and LLM-based axial coding methods. Best values across all methods are shown in bold.

Method (params)	Type	Cov	#Cats	Brev	Coh	Nov	Div
DBSCAN (MiniLM, ms=10)	Clust.	0.122	13	3.25	0.932	0.385	0.712
DBSCAN (MiniLM, ms=20)	Clust.	0.103	6	3.29	**0.953**	0.500	0.708
DBSCAN (mpnet, ms=20)	Clust.	0.113	7	3.00	0.939	0.143	0.700
HDBSCAN (paraphrase+UMAP, mcs=10)	Clust.	0.603	80	3.27	0.690	0.588	0.567
HDBSCAN (MiniLM+UMAP, mcs=20)	Clust.	0.521	49	3.26	0.676	0.490	0.575
HDBSCAN (BGE+UMAP, mcs=50)	Clust.	**0.960**	3	3.05	0.661	0.333	0.526
Deepseek-R1	LLM	0.203	67	2.90	0.478	0.239	**0.134**
Llama-4 Maverick	LLM	0.328	83	2.30	0.576	**0.024**	0.225
Mixtral 8×7B	LLM	0.199	33	**1.80**	0.499	0.121	0.260

and Mixtral 8×7B achieve lower ROUGE-L overall, though BERTScore remains uniformly high and less discriminative across models.

Summary. Overall, clustering methods, in particular HDBSCAN, offer strong coverage and control over granularity but risk summarizing debates into overly broad categories. LLM grouping, on the contrary, produces compact and interpretable labels and achieves the strongest ROUGE scores but leaves a large portion of utterances unassigned. This trade-off between coverage and external alignment is illustrated in Fig. 3, where we plot coverage on the x-axis and similarity on the y-axis. The figure highlights a clear coverage–alignment trade-off between clustering-based and LLM-based axial coding methods. HDBSCAN variants (orange crosses) achieve high coverage, with moderate alignment to the gold taxonomy. LLM-based methods (pink triangles) show the opposite pattern: substantially higher similarity, at the cost of much lower coverage. DBSCAN variants (green) perform comparatively poorly on both dimensions, exhibiting especially low coverage by discarding most utterances.

Data availability. We release the 5k test subset coded by the best clustering configurations and LLMs, including utterance text, human domain/subtopic labels, open codes, and axial categories.[6]

7 Discussion

The two axial coding strategies show complementary strengths. Clustering, particularly HDBSCAN with BGE embeddings and UMAP, achieve near-complete coverage and moderate external alignment, but often at the cost of overly broad categories (for example, only three clusters in the best configuration). DBSCAN

[6] `https://github.com/Likich/axial_coding_dataset`

Table 5. Extrinsic evaluation results, showing ROUGE-L (RL), Cosine similarity (Cos), and BERTScore (BERT) at the Domain and Subtopic level for both clustering and LLM-based axial coding methods. Best values across all methods are in bold.

Method (params)	**Type**	**Domain**			**Subtopic**		
		RL	Cos	BERT	RL	Cos	BERT
DBSCAN (MiniLM, ms=10)	Clust.	0	0.098	0.832	0.008	0.121	0.835
DBSCAN (MiniLM, ms=20)	Clust.	0	0.107	0.828	0	0.143	0.832
DBSCAN (mpnet, ms=20)	Clust.	0	0.113	0.840	0	0.148	0.844
HDBSCAN (paraphrase+UMAP, mcs=10)	Clust.	0.004	0.126	0.835	0.028	0.169	0.842
HDBSCAN (MiniLM+UMAP, mcs=20)	Clust.	0.015	0.138	0.835	0.038	0.183	0.842
HDBSCAN (BGE+UMAP, mcs=50)	Clust.	0	0.179	0.846	0.111	0.300	0.867
Deepseek-R1	LLM	**0.076**	**0.277**	0.853	**0.248**	**0.328**	0.859
Llama-4 Maverick	LLM	0.015	0.215	0.853	0.057	0.252	0.862
Mixtral 8×7B	LLM	0.004	0.207	**0.872**	0.001	0.187	**0.871**

variants discard most of the utterances as noise and thus fail to align with the taxonomy.

LLM-based grouping, on the contrary, produces compact and interpretable categories, with succinct labels (1.8–2.9 words), and lower divergence from the aggregated code space. The best-performing LLM (Deepseek-R1) outperforms the clustering methods in external alignment with subtopics, though with much lower coverage. Together, clustering methods optimize *coverage and structural separation*, while LLMs excel in *label quality and semantic alignment*. A hybrid approach using clustering for structural grouping and LLMs for semantic summarization offers the most balanced trade-off for axial coding of debates. A concrete pipeline would proceed in two stages: (i) apply clustering to ensure full corpus coverage and establish coarse structural partitions, and (ii) apply LLM-based summarization within each cluster to generate concise, interpretable axial labels. This preserves recall while leveraging LLMs where they are strongest—semantic abstraction and naming. We leave empirical validation of this hybrid strategy to future work.

Our results suggest that different axial coding methods may optimize distinct retrieval scenarios and information needs. Clustering methods, with broad coverage and structural separation, provide a strong basis for recall-oriented search, where the goal is to capture all potentially relevant utterances under coarse themes. LLM-based grouping, with concise and interpretable labels, may better support precision-oriented search, surfacing categories that are meaningful and user-friendly for downstream search or recommendation tasks. Thus, a hybrid pipeline could bridge IR needs by combining clustering for exhaustive indexing with LLM summarization for interpretability.

7.1 Limitations and future work

Our study is limited in several respects. First, we focus on a single debate corpus, which constrains the generalizability of findings. Political debates have distinctive structures and rhetorical styles, and it remains to be tested whether our methods extend to other qualitative domains such as interviews, focus groups, or ethnographic texts. Second, we evaluate only a fixed set of LLMs and clustering algorithms; more recent or larger models, alternative prompting strategies, and other clustering approaches could yield different outcomes. Third, while our evaluation covers both extrinsic and intrinsic metrics, it still assumes a relatively clean gold taxonomy; future work should address settings where taxonomies are partial, evolving, or contested. Finally, although we illustrate the potential of hierarchical concept graphs in Fig. 2, our experiments emphasize only the first stage of axial coding. Deeper hierarchies, dynamic graph construction, and applications such as temporal theme tracking, cross-party comparisons, or integration with political information retrieval systems remain to be explored.

8 Conclusion

This paper introduces a method for transforming raw utterances in political debates into structured conceptual representations via open and axial coding with ensemble LLMs. By combining a moderator-based open coding stage with clustering-based grouping and direct LLM-based grouping, we demonstrate how inductive coding can be scaled while retaining interpretability.

Our results reveal complementary strengths. Density-based clustering methods discover cluster counts close to the annotated domains and yield high internal separation, but exclude a substantial portion of utterances as noise in comparison to partition-based methods. However, even if partition-based clustering methods ensure full coverage of utterances, they require fixing the number of clusters in advance, and tend to sacrifice interpretive quality. Direct LLM grouping, on the other hand, while achieving lower coverage than HDBSCAN, adapts flexibly to the data and produces concise, human-readable labels that better support interpretation.

Overall, our experiments show that clustering provides broad structural coverage of debates, while LLM grouping produces compact and semantically meaningful categories. Rather than treating these as competing strategies, we view them as complementary: clustering ensures that a vast number of utterances are represented, and LLMs supply the interpretive layer that makes categories comprehensible.

Acknowledgements. David Graus is partly funded by ICAI (AI for Open Government Lab). Views expressed in this paper are not necessarily shared or endorsed by those funding the research.

Disclosure of Interests The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Azarbonyad, H., Dehghani, M., Kenter, T., Marx, M., Kamps, J., de Rijke, M.: HiTR: Hierarchical Topic Model Re-Estimation for Measuring Topical Diversity of Documents . IEEE Transactions on Knowledge & Data Engineering **31**(11), 2124–2137 (Nov 2019). https://doi.org/10.1109/TKDE.2018.2874246
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**(null), 993–1022 (Mar 2003)
3. Bommasani, R., Hudson, D.A., Adeli, E., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021)
4. Braun, V., Clarke, V.: Thematic Analysis: A Practical Guide. SAGE Publications (2021)
5. Chen, J., Lotsos, A., Cheng, S., Wang, C., Zhao, L., Hullman, J., Sherin, B., Wilensky, U., Horn, M.: A computational method for measuring "open codes" in qualitative analysis (2025), https://arxiv.org/abs/2411.12142
6. Cormack, G.V., Grossman, M.R., Hedin, B., Oard, D.W.: Overview of the trec 2010 legal track. In: TREC (2010)
7. Creswell, J., Báez, J.: 30 Essential Skills for the Qualitative Researcher. SAGE Publications, Incorporated (2021)
8. Fidel, R.: Qualitative methods in information retrieval research. Library and information science research **15**, 219–219 (1993)
9. Fischer, T., Biemann, C.: Exploring large language models for qualitative data analysis. In: Hämäläinen, M., Öhman, E., Miyagawa, S., Alnajjar, K., Bizzoni, Y. (eds.) Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities. pp. 423–437. Association for Computational Linguistics, Miami, USA (Nov 2024). https://doi.org/10.18653/v1/2024.nlp4dh-1.41
10. Glaser, B., Strauss, A.: Discovery of grounded theory: Strategies for qualitative research. Routledge (2017)
11. Grootendorst, M.: Bertopic: Neural topic modeling with a class-based tf-idf procedure (2022), https://arxiv.org/abs/2203.05794
12. Kaptein, R., Marx, M., Kamps, J.: Who said what to whom? capturing the structure of debates. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. pp. 831–832. SIGIR '09, Association for Computing Machinery, New York, NY, USA (Jul 2009). https://doi.org/10.1145/1571941.1572151
13. Laban, P., Schnabel, T., Bennett, P.N., Hearst, M.A.: SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. Transactions of the Association for Computational Linguistics **10**, 163–177 (2022). https://doi.org/10.1162/tacl_a_00453
14. Leeson, W., Resnick, A., Alexander, D., Rovers, J.: Natural language processing (nlp) in qualitative public health research: A proof of concept study. International Journal of Qualitative Methods **18**, 1609406919887021 (2019). https://doi.org/10.1177/1609406919887021
15. Lupu, M., Hanbury, A.: Patent retrieval. Foundations and Trends® in Information Retrieval **7**(1), 1–97 (2013). https://doi.org/10.1561/1500000027
16. Makki, R., Carvalho, E., Soto, A.J., Brooks, S., Oliveira, M.C.F.D., Milios, E., Minghim, R.: ATR-Vis: Visual and Interactive Information Retrieval for Parliamentary Discussions in Twitter. ACM Trans. Knowl. Discov. Data **12**(1), 3:1–3:33 (Feb 2018). https://doi.org/10.1145/3047010

17. Matter, D., Schirmer, M., Grinberg, N., Pfeffer, J.: Close to human-level agreement: Tracing journeys of violent speech in incel posts with gpt-4-enhanced annotations (2024)
18. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.J.: Introduction to WordNet: An On-line Lexical Database*. International Journal of Lexicography **3**(4), 235–244 (dec 1990). https://doi.org/10.1093/ijl/3.4.235
19. Oard, D.W., Webber, W.: Information retrieval for e-discovery. Foundations and Trends® in Information Retrieval **7**(2–3), 99–237 (2013). https://doi.org/10.1561/1500000025
20. Parfenova, A.: Automating the information extraction from semi-structured interview transcripts. In: Companion Proceedings of the ACM Web Conference 2024. pp. 983–986 (2024)
21. Parfenova, A., Denzler, A., Pfeffer, J.: Automating qualitative data analysis with large language models. In: Fu, X., Fleisig, E. (eds.) Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop). pp. 177–185. Association for Computational Linguistics, Bangkok, Thailand (Aug 2024), https://aclanthology.org/2024.acl-srw.17
22. Parfenova, A., Pfeffer, J.: Measuring what matters: Evaluating ensemble LLMs with label refinement in inductive coding. In: Che, W., Nabende, J., Shutova, E., Pilehvar, M.T. (eds.) Findings of the Association for Computational Linguistics: ACL 2025. pp. 10803–10816. Association for Computational Linguistics, Vienna, Austria (Jul 2025). https://doi.org/10.18653/v1/2025.findings-acl.563
23. Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., Manandhar, S.: SemEval-2014 task 4: Aspect based sentiment analysis. In: Nakov, P., Zesch, T. (eds.) Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). pp. 27–35. Association for Computational Linguistics, Dublin, Ireland (Aug 2014). https://doi.org/10.3115/v1/S14-2004
24. de Rooij, O., Odijk, D., de Rijke, M.: ThemeStreams: visualizing the stream of themes discussed in politics. In: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. pp. 1077–1078. SIGIR '13, Association for Computing Machinery, New York, NY, USA (Jul 2013). https://doi.org/10.1145/2484028.2484215
25. Salah, Z., Coenen, F., Grossi, D.: Extracting debate graphs from parliamentary transcripts: a study directed at UK house of commons debates. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law. pp. 121–130. ICAIL '13, Association for Computing Machinery, New York, NY, USA (Jun 2013). https://doi.org/10.1145/2514601.2514615
26. Saldaña, J.: The Coding Manual for Qualitative Researchers. Core textbook, SAGE (2021)
27. Spinoso-Di Piano, C., Rahimi, S., Cheung, J.: Qualitative code suggestion: A human-centric approach to qualitative coding. In: Bouamor, H., Pino, J., Bali, K. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2023. pp. 14887–14909. Association for Computational Linguistics, Singapore (Dec 2023). https://doi.org/10.18653/v1/2023.findings-emnlp.993
28. Törnberg, P.: Large language models outperform expert coders and supervised classifiers at annotating political social media messages. Social Science Computer Review **0**(0), 08944393241286471 (0). https://doi.org/10.1177/08944393241286471
29. Xiao, Z., Yuan, X., Liao, Q.V., Abdelghani, R., Oudeyer, P.Y.: Supporting qualitative analysis with large language models: Combining codebook with gpt-3 for

deductive coding. In: 28th International Conference on Intelligent User Interfaces. IUI '23, ACM (Mar 2023). https://doi.org/10.1145/3581754.3584136
30. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating text generation with BERT. CoRR **abs/1904.09675** (2019), http://arxiv.org/abs/1904.09675
31. Ziems, C., Held, W., Shaikh, O., Chen, J., Zhang, Z., Yang, D.: Can large language models transform computational social science? Computational Linguistics **50**(1), 237–291 (Mar 2024). https://doi.org/10.1162/coli_a_00502