

TaxCE: A Framework for Automated Taxonomy Construction and Evaluation at Scale

Sandeep Sricharan Mukku

Amazon
smukku@amazon.com

Albert Aristotle Nanda

Amazon
alnanda@amazon.com

Rohit Pyati

Amazon
pyarohit@amazon.com

Abstract

Organizing unstructured feedback text into hierarchical taxonomy is a fundamental challenge in NLP, particularly in domains where feedback arrives at massive scale in varied forms such as reviews, transcripts, and surveys. Existing approaches either produce shallow hierarchies, neglect long-tail topics, or lack rigorous evaluation frameworks. We present TaxCE, a fully automated framework that constructs multi-level hierarchical taxonomies from raw text through progressive condensation of corpus content into actionable segments, deduplicated semantic units, and granular topics with definitions, which are then organized bottom-up into a hierarchy with corpus-groundedness. We also introduce three corpus-grounded evaluation metrics, Exclusivity, Exhaustivity, and Granularity (EEG), and integrate them into a metrics-in-the-loop iterative refinement mechanism that diagnoses deficiencies and applies targeted corrections until convergence. Extensive experiments demonstrate that TaxCE consistently outperforms existing baselines spanning classical topic models, neural methods, and LLM-based approaches, with average improvements of 11.8, 20.5, and 15.7 percentage points in exclusivity, exhaustivity, and granularity respectively over the strongest baseline. Human evaluation further confirms superior taxonomy quality, actionability, and navigability.

1 Introduction

The ability to automatically organize large volumes of unstructured feedback into actionable, hierarchical structures is increasingly critical for organizations that rely on customer and stakeholder input for decision-making. Product reviews, chat transcripts, survey responses, and support tickets collectively form rich but unwieldy corpora spanning domains as diverse as healthcare, finance, retail, telecommunications, and public services. A well-constructed

taxonomy, arranging topics¹ from coarse (broad) to granular (specific), enables systematic routing, analysis, insight extraction, and classification at scale. Building such taxonomies has traditionally required domain experts and ontologists to manually identify, scope, and hierarchically organize topics (Bansal et al., 2014; Bordea et al., 2015), a process that is expensive, subjective, and difficult to reproduce. Automated approaches based on topic modeling and taxonomy induction exhibit three critical limitations: (a) **Shallow hierarchies**: most methods produce flat or shallow two-level hierarchies, failing to capture multi-level granularity where a single broad category (e.g., “*Hardware Issues*”) may encompass dozens of actionable sub-topics or granular intents (e.g., “*Fast Battery Drain*”, “*Broken Charging Port*”); (b) **Incomplete coverage**: existing approaches focus on prominent topics while neglecting the long tail of less frequent but important issues, often relegating them to catch-all categories like “*Other*” or “*Miscellaneous*”, violating the Mutually Exclusive, Collectively Exhaustive (MECE) principle (Minto, 1987); and (c) **Lack of rigorous evaluation**: no established framework exists for rigorously evaluating taxonomy quality, with current methods relying on extrinsic proxies or ad hoc human judgments that are expensive and non-reproducible. We argue that taxonomy quality must be assessed along three complementary dimensions, which we collectively term **EEG**: **Exclusivity** ($\mathcal{E}$), the semantic distinctiveness among topics at the same level of the taxonomy tree; **Exhaustivity** ($\mathcal{X}$), the coverage of all actionable intents in the corpus into the taxonomy; and **Granularity** ($\mathcal{G}$), how specific and actionable the leaf topics are in representing the underlying corpus content. No prior work jointly optimizes for all three or provides formal, corpus-grounded definitions for these metrics.

¹We sometimes use the words *topic*, and *node* interchangeably throughout this paper.

To address these gaps, we present **Taxonomy Construction and Evaluation (TaxCE)**, read as *Taxie*, a fully automated framework that takes raw unstructured feedback as input and produces a validated, multi-level hierarchical taxonomy as output. The key idea is to treat taxonomy construction as a knowledge condensation problem: raw documents are first distilled into atomic, deduplicated semantic units (which we term *concepts*), preserving specificity and granularity of the underlying intents. These concepts serve as building blocks from which granular leaf topics, their definitions, and the hierarchical structure are derived bottom-up. TaxCE requires no seed terms, labeled data, or domain ontology, and operates with minimal human supervision. It is language-agnostic, with translation handled at the extraction stage. Our contributions are: (1) **Framework for end-to-end automated taxonomy construction.** We propose TaxCE, introducing *concept generation* and *standardization* as a novel intermediate representation that bridges raw text and taxonomy nodes while ensuring deduplication and traceability. (2) **Corpus-grounded evaluation metrics.** We introduce formal definitions for three complementary taxonomy quality metrics (EEG), explicitly grounded to the input corpus, and integrate them into a *metrics-in-the-loop* iterative refinement mechanism. Additionally, we present the first systematic benchmarking of automated taxonomy generation methods using EEG metrics alongside human evaluation, and demonstrate TaxCE’s generalizability across multiple domains through stage-wise evaluation and ablation studies.

2 Related work

Early taxonomy construction methods extracted hypernym-hyponym relations using lexico-syntactic patterns (Hearst, 1992), distributional similarity (Snow et al., 2004), and structured learning (Bansal et al., 2014), with SemEval shared tasks (Bordea et al., 2015, 2016) establishing term-level benchmarks where systems such as TAXI (Panchenko et al., 2016) combined pattern matching with focused crawling to achieve strong performance. Neural methods advanced this significantly: TaxoGen (Zhang et al., 2018) introduced unsupervised topic taxonomy construction through adaptive term embedding and recursive clustering but is limited to two-level hierarchies; HiExpan (Shen et al., 2018) proposed task-guided

construction via hierarchical tree expansion but requires seed taxonomies; NetTaxo (Shang et al., 2020) leverages text-rich network signals but assumes network structure availability; CoRel (Huang et al., 2020) performs seed-guided construction via concept learning and relation transferring, limiting domain transferability; and Mao et al. (2018) proposed end-to-end reinforcement learning for taxonomy induction but targeted term-level rather than topic-level hierarchies. TaxoCom (Lee et al., 2022) and TaxoEnrich (Jiang et al., 2022) address taxonomy completion and enrichment respectively but assume existing partial taxonomies as input. More recently, Babaei Giglou et al. (2023) explored LLMs for ontology learning, though LLM-generated structures remain prone to hallucination and lack corpus grounding, and Mukku et al. (2023) introduced structured insight mining from customer feedback incorporating a taxonomy component but without full automation or systematic evaluation. Across these works, none jointly produce deep multi-level hierarchies, atomic corpus-grounded concepts, and topic definitions in a fully automated manner from raw text alone. In topic modeling, LDA (Blei et al., 2003) and its hierarchical extension hLDA (Griffiths et al., 2003) discover latent topics but produce unstructured sets without explicit parent-child semantics. Neural topic models including ProLDA (Srivastava and Sutton, 2017) and ETM (Dieng et al., 2020) improve topic quality via variational autoencoders, and embedding-based methods such as BERTopic (Grootendorst, 2022) and TopClus (Meng et al., 2022) leverage pretrained representations for topic discovery, but all produce flat topic sets without hierarchical structure or topic definitions, making them unsuitable for commercial use as they limit navigability.

Taxonomy evaluation remains fragmented, relying on downstream classification accuracy (Shen et al., 2021), structural measures such as edge precision and ancestor F1 (Bordea et al., 2015, 2016; Mao et al., 2018), ad hoc human ratings (Zhang et al., 2018; Huang et al., 2020), or flat topic quality measures such as coherence (Mimno et al., 2011) and diversity (Dieng et al., 2020) that ignore hierarchical structure. These approaches either measure structural properties in isolation, repurpose metrics designed for flat topic sets, or conflate taxonomy quality with downstream classifier quality. No prior work simultaneously measures EEG with corpus-grounded definitions or integrates such metrics into the construction loop.

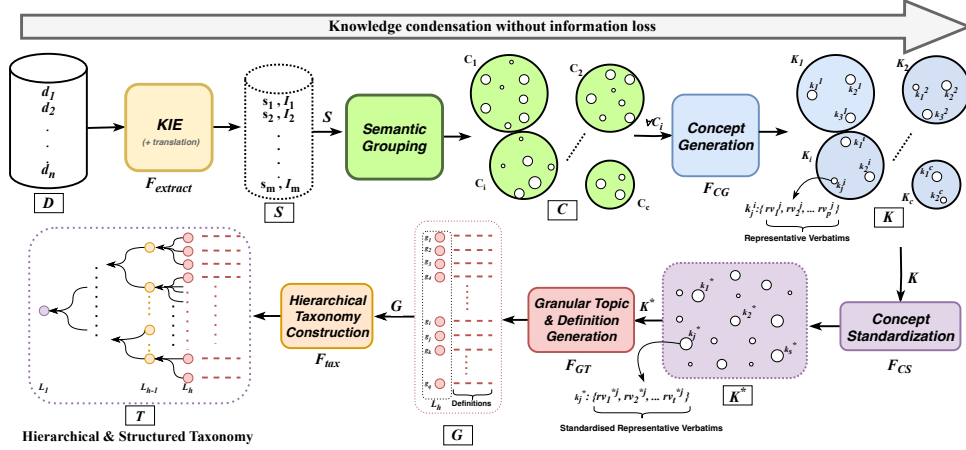


Figure 1: TaxCE framework with progressive knowledge condensation, followed by root-to-leaf validation (F_{val}) and EEG-driven iterative refinement that loops back to grouping until convergence.

3 Problem statement

Given a raw, unstructured corpus $D = \{d_1, d_2, \dots, d_n\}$ with no predefined schema, seed terms, or domain ontology, and a specification of the key information types to be captured in the taxonomy (viz., categories, aspects, root causes, resolutions, etc.), the goal is to automatically construct a multi-level hierarchical taxonomy $\mathcal{T}^*$ with granular, actionable leaf topics hierarchically organized from coarse to specific. The constructed taxonomy must simultaneously satisfy EEG: $\mathcal{E}(\mathcal{T}^*) \geq \theta_E$, $\mathcal{X}(\mathcal{T}^*, D) \geq \theta_X$, and $\mathcal{G}(\mathcal{T}^*, D) \geq \theta_G$, where $\theta_E, \theta_X, \theta_G$ are minimum quality thresholds.

Terminology. An **actionable segment** s is a contiguous text span conveying a self-contained piece of information; an **intent** $\mathcal{I}$ is a semantic label summarizing s derived from the full document context, forming an *intent-tuple* $(s, \mathcal{I})$. A **concept** κ_i is a deduplicated semantic unit grouping semantically same intents, associated with representative verbatims $\{rv_1^i, \dots, rv_p^i\}$ selected for diversity and specificity; the full set is $\mathcal{K}$ and the standardized set is $\mathcal{K}^*$. A **granular topic** g_j is a specific, actionable leaf-level topic derived from one or more concepts in $\mathcal{K}^*$, with a natural language definition; the full set is $\mathcal{G}$. The **taxonomy** $\mathcal{T}$ is a rooted tree of depth h with nodes at level i denoted L_i (L_1 : root, L_h : leaves) and children of node v denoted $\text{Ch}(v) \subseteq L_{i+1}$.

4 Methodology

Framework Overview. TaxCE constructs a hierarchical taxonomy from raw corpus through 6 stages (Figure 1): (1) Key Information Extraction,

(2) Semantic Grouping, (3) Concept Generation and Standardization, (4) Granular Topic and Definition Generation, (5) Hierarchical Taxonomy Construction with Root-to-Leaf Validation, and (6) Iterative Refinement. TaxCE employs a two-tier LLM strategy: Stage 1 uses a cost-efficient, locally-hosted LLM for high-volume extraction (Appendix E), while Stages 3–5 use an instruction-tuned LLM with advanced reasoning and longer context windows, invoked far fewer times. Phase-wise examples illustrating each stage are provided in Appendix P.

4.1 Key information extraction

For each document $d \in D$, TaxCE extracts actionable segments and assigns intent labels jointly using an instruction-tuned LLM (F_{extract}). The types of key information to extract (viz., categories, aspects, root causes, resolutions, polarity, etc.) are specified by the use case, producing intent-tuples $S = \bigcup_{d \in D} F_{\text{extract}}(d) = \{(s_1, \mathcal{I}_1), \dots, (s_m, \mathcal{I}_m)\}$. The intent $\mathcal{I}_i$ enriches the segment s_i with broader feedback-level context, critical for feedback types such as surveys and transcripts where verbatims² alone may lack sufficient context. For multilingual corpora, extraction simultaneously translates segments and intents into the target taxonomy language; all subsequent stages operate in this single language. Intents are used solely to enrich semantic representations during grouping and are not carried forward beyond that stage.

4.2 Semantic grouping

The intent-tuples S are organized into c semantically coherent, non-overlapping groups $C =$

²A *verbatim* refers to the exact text span extracted from the original feedback, preserved without paraphrasing.

$\{C_1, \dots, C_c\}$ such that $\bigcup_{i=1}^c C_i = S$ and $\forall i \neq j, C_i \cap C_j = \emptyset$. Each intent-tuple $(s_i, \mathcal{I}_i)$ is concatenated into $x_i = \text{concat}(s_i, \mathcal{I}_i)$ and encoded using a sentence embedding model (Reimers and Gurevych, 2019) to obtain $e_i = \text{Encode}(x_i)$. Concatenating the intent with the segment enriches the embedding with broader semantic context, improving grouping quality even when segments alone are ambiguous. The framework is agnostic to the grouping algorithm; our implementation uses HDBSCAN (Campello et al., 2013), which automatically determines the number of groups via density estimation without requiring c to be specified a priori.³

4.3 Concept generation and standardization

Concept Generation. Each group C_i may contain multiple granular intents within the same semantic neighborhood that need to be separated into individual concepts. For each group, TaxCE identifies concepts $\mathcal{K}_i = \{\kappa_1^i, \dots, \kappa_t^i\}$ where $1 \leq t \leq t_{\max}$, with complete concept set $\mathcal{K} = \bigcup_{i=1}^c \mathcal{K}_i$. This step uses an instruction-tuned LLM (F_{CG}), which analyzes intent-tuples within each group and identifies distinct concepts. For each concept κ_j^i , F_{CG} selects up to p actionable segments from C_i as representative verbatims $\{rv_1^j, \dots, rv_p^j\}$, where p is determined by F_{CG} based on the diversity of mapped intent-tuples, chosen to cover all variations and describe the concept’s scope. Concepts within each group are encouraged to be semantically distinct, though corpus-level distinctness is enforced during standardization. See Appendix P.3 for illustrative examples.

Concept Standardization. Since grouping is inherently imperfect, semantically related intent-tuples on group boundaries may be assigned to different groups, causing semantically same concepts to emerge independently. Before standardization, representative verbatims across such duplicates may also overlap semantically, introducing redundancy at both concept and verbatim levels. To remove this corpus-wide redundancy, TaxCE performs a lightweight clustering of all concepts in $\mathcal{K}$ based on their embeddings to form batches of semantically related concepts. This batching (a) groups candidate duplicates for efficient comparison and (b) respects

LLM token length limitations. Each batch is processed by an instruction-tuned LLM (F_{CS}), which identifies semantically same concepts, merges them into canonical concepts, and consolidates representative verbatims by removing semantic overlaps. The resulting $\mathcal{K}^* = \{\kappa_1^*, \dots, \kappa_s^*\}$ consists of atomic, highly specific, standardized concepts, each associated with non-overlapping representative verbatims $\{rv_1^{*j}, \dots, rv_t^{*j}\}$ that collectively represent D without information loss, ensuring exhaustivity. See Appendix P.4 for examples.

4.4 Granular topic and definition generation

Using $\mathcal{K}^*$ and its representative verbatims, TaxCE identifies granular topics forming the leaf nodes, using an instruction-tuned LLM (F_{GT}). Concepts in $\mathcal{K}^*$ may be directly suitable as leaf topics or may need merging to form coherent, non-overlapping topics. If $\mathcal{K}^*$ fits within the LLM’s context window, F_{GT} processes them in a single pass; otherwise, lightweight clustering creates batches processed separately. F_{GT} generates $\mathcal{G} = \{g_1, \dots, g_q\}$ where each g_j must be (a) *specific*: a single, well-defined, end-user usable actionable topic, (b) *independent* and *self-contained*: semantically distinct from all other granular topics and carrying complete meaning in itself, and (c) *grounded*: traceable to concepts in $\mathcal{K}^*$ and transitively to segments in D . For each topic, F_{GT} simultaneously generates a natural language definition with representative verbatims as illustrative examples, serving as human-readable descriptions that downstream classifiers use to assign feedback to the correct topic. Each topic is assigned a polarity from $\{\text{positive}, \text{negative}, \text{neutral}\}$, with neutral mostly assigned for inquiry/question intents. Additionally, TaxCE generates catch-all topics (wherever applicable) to capture generic or ambiguous feedback that lacks actionable specifics (e.g., “complete product dissatisfaction”, “general product dislike”), ensuring exhaustive coverage without forcing vague feedback into inappropriate granular topics. Hyperparameter choices for concept and topic generation are discussed in Appendix C.

4.5 Hierarchical taxonomy construction

The granular topics $\mathcal{G}$ form the leaf level L_h . To construct the full hierarchy, TaxCE groups semantically related topics and generates abstract parent topics representing their combined meaning, repeating this bottom-up process until depth h is reached, yielding $\mathcal{T} = F_{\text{tax}}(\mathcal{G}, h) = \{L_h, L_{h-1}, \dots, L_2, L_1\}$, where

³Alternatives such as Spherical K-Means (Dhillon and Modha, 2001) (with c determined via elbow method and silhouette analysis), DBSCAN (Ester et al., 1996), agglomerative clustering (Müllner, 2011), or fast clustering (<https://www.sbert.net/examples/applications/clustering/README.html#fast-clustering>) can be substituted. See Appendix B for a comparative analysis.

F_{tax} is an instruction-tuned LLM that performs hierarchical grouping and parent topic generation. Since topics in $\mathcal{G}$ trace back through $\mathcal{K}^*$ to actionable segments in D , every leaf node is inherently traceable to corpus evidence. Navigability analysis across different depths is in Appendix D.

Root-to-leaf validation. TaxCE validates the taxonomy using an instruction-tuned LLM (F_{val}) by traversing every root-to-leaf path $\pi = (v^{L_1}, v^{L_2}, \dots, v^{L_h})^4$, verifying: (a) *logical coherence*: the progression from abstract to granular represents a semantically meaningful specialization; and (b) *corpus support*: supporting evidence exists in D for the topic combination along the path. Paths failing either condition are flagged for restructuring and, in rare cases, removal.

4.6 Iterative refinement

TaxCE evaluates the constructed taxonomy using EEG metrics (Section 5) and iteratively refines it until all three meet their thresholds $\theta_E, \theta_X, \theta_G$. At each iteration, TaxCE identifies the deficient metric and applies targeted corrections: (a) **Low $\mathcal{E}$** : re-groups with fewer groups ($c' < c$) to consolidate overlapping topics; (b) **Low $\mathcal{G}$** : increases groups ($c'' > c$) to preserve finer-grained concepts; (c) **Low $\mathcal{X}$** : increases groups ($c''' > c$) and reduces standardization aggressiveness. When multiple metrics are deficient, refinement prioritizes $\mathcal{E} \rightarrow \mathcal{G} \rightarrow \mathcal{X}$. Convergence is typically achieved within 1–3 iterations (Appendix F).

5 Evaluation metrics

Evaluating taxonomy quality requires metrics that capture complementary aspects of structure and coverage. We propose three corpus-grounded metrics: Exclusivity ($\mathcal{E}$), Exhaustivity ($\mathcal{X}$), and Granularity ($\mathcal{G}$). Unlike prior approaches that rely on extrinsic proxies or human judgment, these metrics are intrinsic, mathematically defined, and explicitly anchored to the input corpus.

Exclusivity. Exclusivity measures the semantic distinctiveness between leaf topics. Drawing from information-theoretic principles of mutual information and semantic independence (Cover and Thomas, 1999), high exclusivity indicates minimal

⁴Each v^{L_i} is a node from level L_i on path π , with v^{L_1} being the root-level topic and v^{L_h} the leaf. Since the taxonomy is a tree, each leaf has exactly one path, so the total number of paths equals $|\mathcal{L}_h|$.

semantic overlap, ensuring each leaf topic occupies a unique region of the embedding space. Let $T = \{t_1, t_2, \dots, t_n\}$ be the set of n leaf topics, each represented by its embedding vector $\mathbf{v}_i$. Exclusivity is defined as:

$$\mathcal{E} = 100 \cdot \left(1 - \frac{1}{\binom{n}{2}} \sum_{i=1}^n \sum_{j=i+1}^n \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \cdot \|\mathbf{v}_j\|} \right) \quad (1)$$

where $\binom{n}{2} = \frac{n(n-1)}{2}$ is the number of unique pairs. A score of 100 indicates perfect distinctiveness (zero average pairwise similarity), while lower scores indicate increasing overlap. Evaluating exclusivity at the leaf level implicitly ensures exclusivity at higher levels, since parent topics are abstractions over disjoint sets of distinct children.⁵

Exhaustivity. Exhaustivity measures the completeness of a taxonomy in covering all actionable intents present in the corpus. Unlike domain-level coverage, this metric is explicitly grounded to D : a taxonomy is exhaustive if it covers every topic actually evidenced in the data. We use the KIE-extracted actionable segments S (see Section 4.1) as the reference set and the leaf topics $T = \{t_1, t_2, \dots, t_n\}$ as the taxonomy’s coverage. Since S is derived directly from the raw corpus D via a single extraction pass that is independent of any downstream taxonomy construction method, it serves as a method-agnostic corpus representation. A segment $s_i \in S$ is considered covered if its cosine similarity to at least one leaf topic exceeds a similarity threshold τ_x (set to 0.65 in all experiments; sensitivity analysis in Appendix I).⁶ Exhaustivity is defined as:

$$\mathcal{X} = 100 \cdot \frac{|\{s \in S : \exists t \in T, \text{sim}(\mathbf{v}_s, \mathbf{v}_t) \geq \tau_x\}|}{|S|} \quad (2)$$

where $\mathbf{v}_a$ denotes the embedding of entity a . A score of 100 indicates every corpus-derived actionable segment maps to at least one leaf topic.

Granularity. Granularity quantifies how specific and actionable the leaf topics are in representing the underlying corpus content. While exhaustivity measures whether segments are covered (binary), granularity measures how closely they are covered

⁵This follows from the hierarchical construction where each parent groups semantically related but distinct children. If children under different parents are distinct, the parents themselves must also be distinct.

⁶An LLM-as-judge approach can alternatively determine coverage, offering higher reliability. However, this requires one LLM call per segment and is practical only when both the corpus and taxonomy are small enough to fit within a single prompt. For larger settings, the cost and prompt engineering effort approach that of building a full classification system, making it a low priority option in general.

Type	Method	Flipkart			CFPB			AskUbuntu		
		$\mathcal{E}$	$\mathcal{X}$	$\mathcal{G}$	$\mathcal{E}$	$\mathcal{X}$	$\mathcal{G}$	$\mathcal{E}$	$\mathcal{X}$	$\mathcal{G}$
Topic	LDA	78.2	31.5	38.4	80.1	33.7	40.2	79.5	28.6	37.8
	hLDA	72.4	38.2	42.7	74.6	40.1	44.5	73.2	36.4	41.9
	BERTopic	81.3	44.6	51.2	82.5	46.8	53.1	80.9	41.3	50.4
	TopClus	79.8	42.3	48.9	81.2	44.1	50.8	78.6	39.5	48.1
LLM	LLM Zero-shot	68.5	52.1	55.3	70.3	54.6	57.2	67.8	48.2	54.1
	LLM + Clustering	74.1	58.4	60.7	76.2	60.3	62.5	73.5	54.7	59.8
TaxCE		85.7	78.3	76.1	87.1	80.5	78.2	86.3	76.9	75.6

Table 1: Exclusivity ($\mathcal{E}$), Exhaustivity ($\mathcal{X}$), and Granularity ($\mathcal{G}$) across all datasets. $\mathcal{X}$ and $\mathcal{G}$ are computed against the shared reference S for all methods.

(continuous). A taxonomy with broad leaf topics may achieve high exhaustivity by loosely covering many segments, but low granularity because the topics are insufficiently specific. Using S as the reference, we compute for each segment $s_i \in S$ the cosine similarity to its best-matching leaf topic:

$$\mathcal{G} = 100 \cdot \frac{1}{|S|} \sum_{i=1}^{|S|} \max_{t \in T} \frac{\mathbf{v}_{s_i} \cdot \mathbf{v}_t}{\|\mathbf{v}_{s_i}\| \cdot \|\mathbf{v}_t\|} \quad (3)$$

where $\mathbf{v}_a$ denotes the embedding of entity a . A high score indicates leaf topics are specific enough to closely represent each segment, while a low score indicates overly broad coverage (see Appendix P for an illustrative example).

6 Experiments

6.1 Setup

Datasets. We evaluate TaxCE on three public datasets: **Flipkart Product Reviews** (Thummar and Vaghani, 2023) (~194K reviews across 104 categories, to demonstrate TaxCE’s language-agnostic capability, 20% of reviews were machine-translated into four additional languages (DE, IT, FR, ES) with 5% each, yielding a multilingual corpus across 5 languages), **CFPB Consumer Complaints**⁷ (~4M financial complaints), and **AskUbuntu** (Lei et al., 2016) (~167K technical support questions).

Baselines. We compare against six methods: LDA, hLDA, BERTopic, TopClus, LLM Zero-shot, and LLM + Clustering. Setup details are in Appendix H.

Evaluation protocol. Exhaustivity and granularity are computed for all methods against the same reference set S : actionable segments extracted

by TaxCE’s KIE stage via a single pass independent of downstream taxonomy construction, ensuring method-agnostic evaluation. This design also guards against evaluation circularity. Although the reference set S is produced by TaxCE’s KIE stage, this is a single upstream extraction pass that operates independently of any downstream taxonomy construction: KIE neither observes the taxonomy nor is it optimised jointly with it. The same S is used to score all baselines identically, and no downstream TaxCE stage (concept generation, standardisation, topic generation, or refinement) has access to S during training or inference. Any advantage TaxCE derives from this evaluation would therefore have to come from producing leaf topics that align more closely with corpus-derived segments than competing methods do, which is precisely the property we want to measure.

Implementation. TaxCE’s KIE stage uses a locally-hosted LLM; downstream stages use an instruction-tuned LLM with advanced reasoning. Semantic grouping uses HDBSCAN; default depth $h = 3$. Full details on LLM selection, grouping algorithms, hyperparameters, and prompts are in Appendices B, C, E, H, O.

6.2 Main results

TaxCE achieves the highest scores across all three metrics on all three datasets (Table 1), with average improvements of 11.8, 20.5, and 15.7 percentage points in exclusivity, exhaustivity, and granularity respectively over the strongest baseline (LLM + Clustering). The gains are most pronounced on exhaustivity and granularity, where the concept generation and standardization pipeline provides the largest advantage.

Flat topic models (LDA, BERTopic, TopClus) achieve reasonable exclusivity (76–83) but score poorly on exhaustivity (28–47) and granularity (36–

⁷<https://www.consumerfinance.gov/data-research/consumer-complaints/>

53), as they capture only prominent topics and represent them as word distributions rather than actionable definitions. LLM-based methods show the opposite trade-off: LLM Zero-shot achieves moderate exhaustivity (48–55) but low exclusivity (66–70) due to overlapping topics generated without corpus-grounded deduplication; LLM + Clustering improves by grounding in corpus groups (exclusivity 72–76, exhaustivity 55–60) but still suffers from within-group redundancy and cross-group overlap. TaxCE addresses both issues through concept generation (condensing groups into atomic semantic units) and standardization (eliminating cross-group redundancy), with iterative refinement closing remaining gaps. Detailed per-method analysis is in Appendix H.

Cross-domain consistency. TaxCE performs consistently across all three domains despite variation in scale (167K to 4M), language (5 languages in Flipkart vs. English-only elsewhere), and domain complexity. CFPB yields the highest scores, likely due to its structured complaint language; AskUbuntu, the smallest corpus, achieves comparable performance, indicating robustness to corpus size. A stage-wise evaluation is provided in Appendix G.

6.3 Ablation study

We isolate the contributions of concept standardization and iterative refinement by removing each from TaxCE (Table 2(a), averaged across all three datasets). All variants are evaluated against the same shared reference S . Removing concept standardization causes the largest drop in exclusivity (−14.4 points), as redundant concepts from different groups propagate to the leaf topics, creating semantic overlap. Exhaustivity slightly increases (+1.3) because no concepts are merged, but this comes at the cost of a cluttered taxonomy. Removing iterative refinement degrades all three metrics, with exhaustivity dropping most (−7.6) since the refinement loop’s primary correction for low coverage is no longer applied. Removing both components yields the lowest scores, confirming that standardization and refinement address complementary quality dimensions.

6.4 Human evaluation

Three annotators independently rate taxonomies from TaxCE, LLM + Clustering, and BERTopic on three criteria (5-point Likert (Likert, 1932)

Variant	$\mathcal{E}$	$\mathcal{X}$	$\mathcal{G}$
TaxCE (Full)	85.8	77.9	76.1
w/o Std.	71.4 $\downarrow 14.4$	79.2 $\uparrow 1.3$	73.8 $\downarrow 2.3$
w/o Iter.Ref.	80.1 $\downarrow 5.7$	70.3 $\downarrow 7.6$	71.5 $\downarrow 4.6$
w/o Both	67.2 $\downarrow 18.6$	71.8 $\downarrow 6.1$	69.4 $\downarrow 6.7$

(a) Ablation study

Method	Qual.	Act.	Nav.
BERTopic	2.8	2.3	1.9
LLM + Clust.	3.5	3.1	3.3
TaxCE	4.4	4.2	4.3
κ	0.71	0.68	0.74

(b) Human evaluation (5-pt Likert)

Table 2: (a) Ablation study. (b) Human evaluation. κ = Fleiss’ kappa (Fleiss, 1971). Per-dataset ablation in Appendix A.

scale): *Taxonomy Quality* (coherence and correctness), *Actionability* (specificity of leaf topics for decision-making), and *Navigability* (ease of coarse-to-granular navigation) and reported in Table 2(b). TaxCE receives the highest ratings across all dimensions, with particularly strong gains on actionability (+1.1 over LLM + Clustering), reflecting the benefit of concept-grounded topic generation that produces specific, well-defined leaf topics. BERTopic scores lowest on navigability (1.9) because it produces flat topic sets without hierarchical structure. Inter-annotator agreement (Fleiss’ κ = 0.68–0.74) indicates substantial agreement.

7 Conclusion

We presented TaxCE, a framework that treats taxonomy construction as progressive knowledge condensation, introducing concept generation and standardization as intermediate representations between raw text and taxonomy nodes. The proposed EEG metrics provide the first unified, corpus-grounded evaluation of exclusivity, exhaustivity, and granularity, and their integration into an iterative refinement loop enables self-correcting taxonomy construction. Results across three domains confirm consistent improvements over all baselines on both automatic and human evaluations. Future work includes analyzing LLM sensitivity across pipeline stages, developing subtree-level refinement, extending EEG to intermediate hierarchical levels, and replacing the current metric-driven correction rules with a learned refinement policy that selects corrective actions from observed EEG trajectories.

Limitations

While TaxCE performs well across three diverse domains, a few limitations are worth noting. Its multi-stage design introduces cascading sensitivity, since suboptimal outputs at early stages such as overly broad concept generation can propagate downstream; iterative refinement helps mitigate this, but careful prompt engineering at each stage still matters. The EEG metrics also depend on cosine similarity in embedding space, so absolute scores are sensitive to the choice of embedding model. Relative rankings across methods stay consistent and cross-embedding EEG correlation lies between 0.89 and 0.97 (Appendix K), though practitioners should recalibrate τ_x when switching backends. The framework supports arbitrary taxonomy depth h but lacks a mechanism to automatically determine the optimal depth; in our experiments $h = 3$ worked best consistently, though other corpora may prefer different values.

On the cost and scope side, the KIE stage processes each document individually to ensure exhaustive coverage, which can be expensive for very large corpora, and while representative sampling reduces cost it risks missing long-tail topics. TaxCE is designed to be language-agnostic via translation at the KIE stage, but our experiments rely primarily on English corpora with only partial multilingual augmentation, and validation on fully non-English corpora was not conducted. Three complementary evaluations also remain open: (i) alignment against CFPB’s official taxonomy using edge precision, ancestor F1, and Wu-Palmer similarity (Wu and Palmer, 1994), (ii) a KIE-free reference-set control using an independent extractor to further isolate the reference S from TaxCE, which the anti-circularity argument in Section 6.1 partially addresses, and (iii) a controlled noise-injection study on KIE outputs to quantify robustness to upstream extraction errors. The corpus-grounded EEG evaluation, cross-domain consistency, and anti-circularity design already support the paper’s core claims, and we consider these additional evaluations natural next steps.

Ethical Considerations

All datasets used in this work are publicly available and contain no personally identifiable information. We acknowledge the following ethical considerations:

1. LLM-generated taxonomies may inherit biases

present in the underlying language model, potentially manifesting as culturally skewed category names or underrepresentation of minority viewpoints. Practitioners deploying TaxCE in production should review generated taxonomies for such biases, particularly when the taxonomy informs downstream decision-making such as complaint routing or content moderation.

2. The work primarily focuses on English-language corpora (with partial multilingual augmentation), which may limit generalizability of findings to other linguistic contexts. We acknowledge this as a scope constraint and encourage future validation across diverse languages.
3. The iterative nature of TaxCE’s pipeline involves multiple LLM inference calls across stages, which carries computational cost and associated environmental impact. We mitigate this by employing cost-efficient locally-hosted models for the high-volume KIE stage and limiting downstream LLM invocations to concept-level (rather than document-level) processing.

References

- Dogu Araci. 2019. [FinBERT: Financial sentiment analysis with pre-trained language models](#). *arXiv preprint arXiv:1908.10063*.
- Hamed Babaei Giglou, Jennifer D’Souza, and Sören Auer. 2023. [Llms4ol: Large language models for ontology learning](#). In *International Semantic Web Conference*, pages 408–427. Springer.
- Mohit Bansal, David Burkett, Gerard De Melo, and Dan Klein. 2014. [Structured learning for taxonomy induction with belief propagation](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1041–1051.
- David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. [Latent dirichlet allocation](#). *Journal of Machine Learning Research*, 3(Jan):993–1022.
- Georgeta Bordea, Paul Buitelaar, Stefano Faralli, and Roberto Navigli. 2015. [Semeval-2015 task 17: Taxonomy extraction evaluation \(TExEval\)](#). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 902–910.
- Georgeta Bordea, Els Lefever, and Paul Buitelaar. 2016. [Semeval-2016 task 13: Taxonomy extraction evaluation \(TExEval-2\)](#). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1081–1091.

- Tadeusz Caliński and Jerzy Harabasz. 1974. [A dendrite method for cluster analysis](#). *Communications in Statistics – Theory and Methods*, 3(1):1–27.
- Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. 2013. [Density-based clustering based on hierarchical density estimates](#). In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 160–172.
- Thomas M Cover and Joy A Thomas. 1999. *Elements of Information Theory*, 2nd edition. John Wiley & Sons.
- David L Davies and Donald W Bouldin. 1979. [A cluster separation measure](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.
- Inderjit S Dhillon and Dharmendra S Modha. 2001. [Concept decompositions for large sparse text data using clustering](#). *Machine Learning*, 42(1):143–175.
- Adji B Dieng, Francisco JR Ruiz, and David M Blei. 2020. [Topic modeling in embedding spaces](#). *Transactions of the Association for Computational Linguistics*, 8:439–453.
- Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. [A density-based algorithm for discovering clusters in large spatial databases with noise](#). In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231.
- Joseph L Fleiss. 1971. [Measuring nominal scale agreement among many raters](#). *Psychological Bulletin*, 76(5):378–382.
- Thomas Griffiths, Michael Jordan, Joshua Tenenbaum, and David Blei. 2003. [Hierarchical topic models and the nested Chinese restaurant process](#). In *Advances in Neural Information Processing Systems*, volume 16, pages 17–24.
- Maarten Grootendorst. 2022. [Bertopic: Neural topic modeling with a class-based tf-idf procedure](#). *arXiv preprint arXiv:2203.05794*.
- Marti A Hearst. 1992. [Automatic acquisition of hyponyms from large text corpora](#). In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*, pages 539–545.
- Jiaxin Huang, Yiqing Xie, Yu Meng, Yunyi Zhang, and Jiawei Han. 2020. [Corel: Seed-guided topical taxonomy construction by concept learning and relation transferring](#). In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1928–1936.
- Minhao Jiang, Xiangchen Song, Jieyu Zhang, and Jiawei Han. 2022. [Taxoerich: Self-supervised taxonomy completion via structure-semantic representations](#). In *Proceedings of the ACM Web Conference 2022*, pages 925–934.
- Dongha Lee, Jiaming Shen, SeongKu Kang, Susik Yoon, Jiawei Han, and Hwanjo Yu. 2022. [Taxocom: Topic taxonomy completion with hierarchical discovery of novel topic clusters](#). In *Proceedings of the ACM Web Conference 2022*, pages 2819–2829.
- Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez. 2016. [Semi-supervised question retrieval with gated convolutions](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1279–1289.
- Rensis Likert. 1932. [A technique for the measurement of attitudes](#). *Archives of Psychology*, 22(140):1–55.
- Yuning Mao, Xiang Ren, Jiaming Shen, Xiaotao Gu, and Jiawei Han. 2018. [End-to-end reinforcement learning for automatic taxonomy induction](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2462–2472.
- Yu Meng, Yunyi Zhang, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. [Topic discovery via latent space clustering of pretrained language model representations](#). In *Proceedings of the ACM Web Conference 2022*, pages 3143–3152.
- David Mimno, Hanna Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. [Optimizing semantic coherence in topic models](#). In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 262–272.
- Barbara Minto. 1987. *The Minto Pyramid Principle*. Minto International, Incorporated.
- Sandeep Sricharan Mukku, Manan Soni, Chetan Agarwal, Jitenkumar Rana, Promod Yenigalla, Rashmi Patange, and Shyam Mohan. 2023. [Insightnet: Structured insight mining from customer feedback](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 552–566.
- Daniel Müllner. 2011. [Modern hierarchical, agglomerative clustering algorithms](#). *arXiv preprint arXiv:1109.2378*.
- OpenAI. 2024. [New embedding models and API updates](#).
- Alexander Panchenko, Stefano Faralli, Eugen Ruppert, Steffen Remus, Hubert Naets, Cédric Fairon, Simone Paolo Ponzetto, and Chris Biemann. 2016. [TAXI at SemEval-2016 task 13: A taxonomy induction method based on lexico-syntactic patterns, substrings and focused crawling](#). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1320–1327.

- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.
- Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. [Exploring the space of topic coherence measures](#). In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 399–408.
- Peter J Rousseeuw. 1987. [Silhouettes: A graphical aid to the interpretation and validation of cluster analysis](#). *Journal of Computational and Applied Mathematics*, 20:53–65.
- Jingbo Shang, Xinyang Zhang, Liyuan Liu, Sha Li, and Jiawei Han. 2020. [Nettaxo: Automated topic taxonomy construction from text-rich network](#). In *Proceedings of The Web Conference 2020*, pages 1908–1919.
- Jiaming Shen, Wenda Qiu, Yu Meng, Jingbo Shang, Xiang Ren, and Jiawei Han. 2021. [Taxoclass: Hierarchical multi-label text classification using only class names](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4239–4249.
- Jiaming Shen, Zeqiu Wu, Dongming Lei, Chao Zhang, Xiang Ren, Michelle T Vanni, Brian M Sadler, and Jiawei Han. 2018. [Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion](#). In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2180–2189.
- Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2004. [Learning syntactic patterns for automatic hypernym discovery](#). In *Advances in Neural Information Processing Systems*, volume 17, pages 1297–1304.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. [MPNet: Masked and permuted pre-training for language understanding](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867.
- Akash Srivastava and Charles Sutton. 2017. [Autoencoding variational inference for topic models](#). *arXiv preprint arXiv:1703.01488*.
- Mansi Thummar and Nirali Vaghani. 2023. [Flipkart products review dataset](#).
- Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. [Text embeddings by weakly-supervised contrastive pre-training](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11897–11916.
- Zhibiao Wu and Martha Palmer. 1994. [Verbs semantics and lexical selection](#). In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138.
- Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muenighoff, Defu Lian, and Jian-Yun Nie. 2024. [C-Pack: Packaged resources to advance general Chinese embedding](#). In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 641–649.
- Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian Sadler, Michelle Vanni, and Jiawei Han. 2018. [Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering](#). In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2701–2709.

A Per-dataset ablation results

We disaggregate the ablation results from Section 6.3 across all three datasets in Table 3. Removing standardization degrades exclusivity most on AskUbuntu (−15.3), where overlapping technical topics produce the most cross-group concept redundancy that standardization resolves. Exhaustivity marginally increases without standardization across all datasets since no concepts are merged, but at the cost of overlapping leaf topics. Removing iterative refinement impacts Flipkart exhaustivity most (−8.2), as the first-pass taxonomy misses long-tail product feedback that the refinement loop recovers. On Flipkart, removing refinement causes the largest granularity drop (−4.9), reflecting the need for finer separation among diverse e-commerce product topics. Removing both components yields the lowest scores across all datasets, with combined degradation exceeding the sum of individual removals on exclusivity (e.g., −18.9 on Flipkart vs. −14.5 and −5.4 individually), confirming synergistic interaction between the two components.

Variant	Flipkart			CFPB			AskUbuntu		
	$\mathcal{E}$	$\mathcal{X}$	$\mathcal{G}$	$\mathcal{E}$	$\mathcal{X}$	$\mathcal{G}$	$\mathcal{E}$	$\mathcal{X}$	$\mathcal{G}$
TaxCE (Full)	85.7	78.3	76.1	87.1	80.5	78.2	86.3	76.9	75.6
w/o Std.	71.2	79.5	73.5	73.6	81.2	75.8	71.0	78.8	73.6
w/o Iter.Ref.	80.3	70.1	71.2	82.1	72.8	73.5	79.4	69.8	71.4
w/o Both	66.8	71.5	69.1	69.1	73.4	71.2	67.3	72.1	69.5

Table 3: Per-dataset ablation results. w/o Std.: without concept standardization; w/o Iter.Ref.: without iterative refinement; w/o Both: without either component.

B Grouping algorithm analysis

We compare four grouping algorithms for the semantic grouping stage (Section 4.2), evaluating both intrinsic grouping quality (silhouette score) and downstream taxonomy quality (EEG metrics averaged across all three datasets) in Table 4. HDBSCAN achieves the highest silhouette score and best downstream EEG performance. Its density-based approach automatically determines the number of groups and adapts to varying group densities, which is particularly beneficial for feedback corpora with long-tailed topic distributions. Spherical K-Means performs comparably when c (the number of groups, which must be specified a priori) is well-tuned via elbow and silhouette analysis, but the additional hyperparameter reduces generalizability across datasets. DBSCAN automatically determines group count but is sensitive to ϵ (the

neighborhood radius that defines density reachability), leading to inconsistent performance across datasets with different embedding density profiles. Agglomerative clustering produces uneven group sizes, with some very large groups and many singletons, complicating downstream concept generation. HDBSCAN noise points (intent-tuples not assigned to any group) are assigned to their nearest group via embedding similarity in a post-processing step.

Algorithm	Auto c	Sil.	$\mathcal{E}$	$\mathcal{X}$	$\mathcal{G}$
HDBSCAN	✓	0.34	85.8	77.9	76.1
Spherical K-Means	✗	0.32	84.5	76.8	75.1
DBSCAN	✓	0.29	83.1	74.2	73.5
Agglomerative	✗	0.27	81.5	73.8	72.1

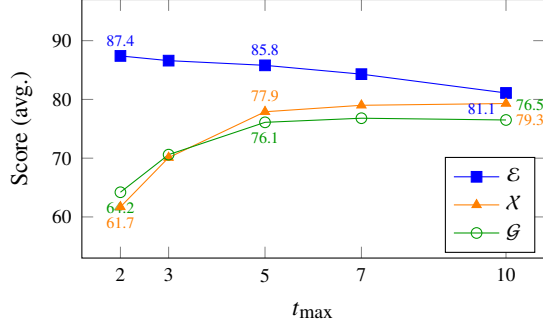
Table 4: Grouping algorithm comparison (averaged across all datasets).

C Hyperparameter analysis

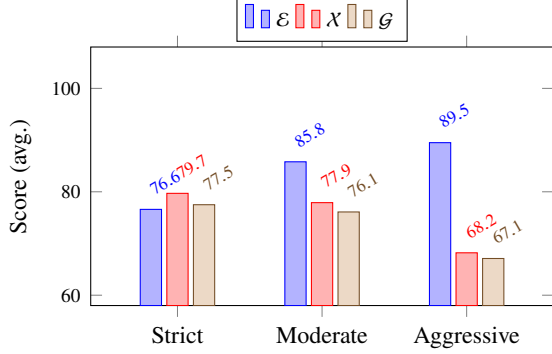
We analyze two key hyperparameters: $t_{\max}$ (maximum concepts per group) and standardization aggressiveness. Figure 2 shows averaged trends; Tables 5 and 6 provide per-dataset breakdowns.

Maximum concepts per group ($t_{\max}$). As shown in Figure 2(a), low $t_{\max}$ constrains concept generation: at $t_{\max} = 2$, groups containing three or more distinct intents are forced to merge, inflating exclusivity (87.4) but severely degrading exhaustivity (61.7) and granularity (64.2). At $t_{\max} = 5$, all three metrics are well-balanced, as most groups contain 2–4 distinct intents in practice. Beyond $t_{\max} = 7$, additional concepts are predominantly near-duplicates that the standardization step must merge, adding computational overhead while exclusivity drops (81.1 at $t_{\max} = 10$) as some fine-grained duplicates survive standardization. The pattern is consistent across all three datasets (Table 5).

Standardization aggressiveness. Figure 2(b) compares three configurations: *strict* (merge only near-identical concepts), *moderate* (merge concepts describing the same issue with different phrasing), and *aggressive* (merge closely related concepts even if they capture different nuances). *Strict* standardization leaves substantial redundancy, degrading exclusivity (76.6) while preserving exhaustivity (79.7). *Aggressive* standardization over-merges, conflating genuinely distinct concepts and degrading exhaustivity (68.2) and granularity (67.1) despite high exclusivity (89.5). *Moderate* achieves the best overall balance. During iterative refinement



(a) Effect of $t_{\max}$ (avg. across datasets).



(b) Effect of standardization aggressiveness.

Figure 2: Hyperparameter sensitivity. (a) $t_{\max} = 5$ balances all three metrics. (b) *Moderate* standardization achieves the best balance.

$t_{\max}$	Flipkart			CFPB			AskUb.		
	$\mathcal{E}$	$\mathcal{X}$	$\mathcal{G}$	$\mathcal{E}$	$\mathcal{X}$	$\mathcal{G}$	$\mathcal{E}$	$\mathcal{X}$	$\mathcal{G}$
2	87.3	62.1	64.5	88.5	64.3	66.2	87.8	60.5	63.8
3	86.5	70.4	70.8	87.8	72.6	72.5	87.1	69.1	70.2
5	85.7	78.3	76.1	87.1	80.5	78.2	86.3	76.9	75.6
7	84.1	79.5	76.8	85.6	81.2	78.9	84.8	78.2	76.3
10	81.2	79.8	76.5	82.8	81.5	78.6	81.9	78.5	76.1

Table 5: Per-dataset $t_{\max}$ results.

(Section 4.6), TaxCE dynamically shifts toward *strict* when exhaustivity is low and toward *aggressive* when exclusivity is low. Per-dataset results in Table 6 confirm this pattern holds across all domains.

D Taxonomy depth analysis

Since TaxCE constructs taxonomies bottom-up (Section 4.5), the granular leaf topics $\mathcal{G}$ are fixed regardless of depth h . Consequently, EEG metrics (which evaluate leaf-level properties) remain stable across depths. The choice of h instead affects *navigability*: the number of intermediate abstraction levels between root and leaf topics. Table 7 reports taxonomy sizes at the default $h = 3$.

At $h = 2$, the first grouping above the leaf level produces hinge-level topics as roots (e.g., 120+ root

Config	Flipkart			CFPB			AskUb.		
	$\mathcal{E}$	$\mathcal{X}$	$\mathcal{G}$	$\mathcal{E}$	$\mathcal{X}$	$\mathcal{G}$	$\mathcal{E}$	$\mathcal{X}$	$\mathcal{G}$
Strict	76.3	80.1	77.5	78.2	82.3	79.4	76.9	78.8	77.1
Moderate	85.7	78.3	76.1	87.1	80.5	78.2	86.3	76.9	75.6
Aggressive	89.4	68.5	67.2	90.8	70.8	69.1	89.7	67.4	66.8

Table 6: Per-dataset standardization results.

Dataset	$ L_1 $	$ L_2 $	$ L_3 $	Avg. $ L_3 $ per L_2
Flipkart	18	120+	1800+	~ 15
CFPB	12	65	450+	~ 6.9
AskUbuntu	8	35	180+	~ 5.1

Table 7: Taxonomy sizes at default depth $h = 3$. $|L_i|$: number of nodes at level i . L1: coarse topics, L2: hinge topics, L3: granular topics.

categories for Flipkart, each containing ~ 15 leaf topics). While individual groups are manageable, the user must navigate 120+ root-level entry points with no higher-level organization, making it difficult to locate relevant topics without prior knowledge of the taxonomy structure. At $h = 3$, a second grouping step produces coarse topics (L1) that organize hinge topics (L2) into a small number of semantically coherent top-level categories (e.g., 18 for Flipkart), providing a natural coarse $\rightarrow$ hinge $\rightarrow$ granular navigation path. Human evaluators consistently rated $h = 3$ taxonomies highest on navigability (Section 6.4). At $h = 4$, an additional grouping above L1 introduces a super-category level that often lacks clear semantic boundaries, adding navigation overhead without meaningful differentiation. At $h = 5$, the hierarchy becomes over-layered with unnecessarily broad abstract categories that do not add informational value, as the semantic gap between consecutive levels becomes too narrow to justify the additional navigation step.

Taxonomy size scales with corpus diversity and scale: Flipkart (104 product categories, multilingual, $\sim 194K$ reviews) produces the largest taxonomy, while AskUbuntu (single technical domain, $\sim 167K$ questions) produces the most compact. The average branching factor from L2 to L3 is notably higher for Flipkart (~ 15) due to the breadth of e-commerce product feedback, compared to 5–7 for the other datasets where domain-specific topics are more tightly scoped.

E LLM selection analysis

TaxCE’s two-tier LLM strategy (Section 4) requires selecting models for the high-volume KIE stage (F_{extract}) and the reasoning-intensive downstream

stages (F_{CG} , F_{CS} , F_{GT} , F_{tax} , F_{val}). Figure 3 reports results for both tiers. For KIE evaluation, we measure F1 against a manually labeled sample of 200 feedback texts⁸ per dataset, annotated by two annotators with disagreements resolved by discussion.

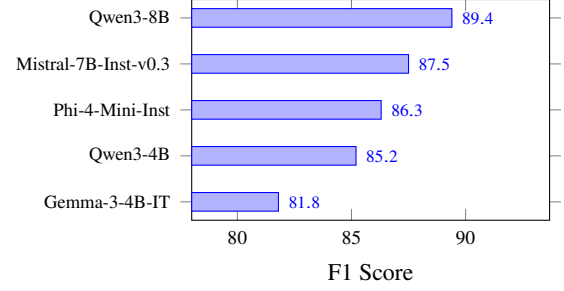
KIE stage. We evaluate five locally-hosted instruction-tuned models under 9B parameters on actionable segment extraction (Figure 3(a)). F1 scales with model size: Qwen3-8B achieves the highest F1 (89.4) but at less than half the throughput of Gemma-3-4B-IT (15.8 vs. 40.1 docs/s). The 3–4B models (Qwen3-4B at 85.2 F1, Phi-4-Mini-Instruct at 86.3 F1) offer the strongest cost-quality balance for corpora exceeding 1M documents, achieving $F1 > 85$ at 30–36 docs/s.

Downstream stages. We evaluate five instruction-tuned LLMs on end-to-end taxonomy quality (Figure 3(b), EEG averaged across all datasets). Claude-3.5-Sonnet and Claude-3-Opus lead with marginal differences (<0.5 points across all metrics), while GPT-4o trails by ~ 1 point. Open-source models (Qwen3-235B-A22B, a mixture-of-experts model with 22B active parameters, and Qwen-Max) trail by 2–4 points, primarily on exhaustivity and granularity, indicating that concept standardization and topic generation benefit from stronger instruction-following capabilities. Given the marginal gap between Sonnet and Opus, Claude-3.5-Sonnet offers the most practical trade-off of quality, cost, and inference speed.

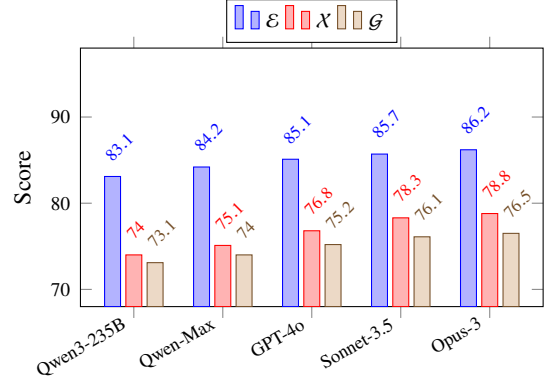
F Iterative refinement convergence

Refinement mechanism details. During refinement, exhaustivity and granularity are computed internally against $\mathcal{K}^*$ to diagnose pipeline-specific deficiencies; the cross-method evaluation against S described in Section 6.1 is used only for final reporting. For low $\mathcal{E}$, TaxCE re-groups with fewer groups, regenerates concepts, re-standardizes, and reconstructs the taxonomy. For low $\mathcal{G}$, TaxCE adjusts standardization to preserve finer-grained concepts that were previously over-merged. For low $\mathcal{X}$, TaxCE reduces standardization aggressiveness to retain concepts that were previously merged away. The priority order ($\mathcal{E} \rightarrow \mathcal{G} \rightarrow \mathcal{X}$) reflects that improving exclusivity can affect the other two metrics and should be stabilized first. Threshold selection guidelines are in Appendix J.

⁸Each feedback text corresponds to a single document $d \in D$ (e.g., one product review, one survey response, or one chat transcript etc.).



(a) KIE LLM: F1 score. Throughput (docs/s): Gemma-3-4B 40.1, Qwen3-4B 36.2, Phi-4-Mini 30.5, Mistral-7B 18.4, Qwen3-8B 15.8.



(b) Downstream LLM comparison (avg. across datasets).

Figure 3: LLM selection analysis. (a) KIE stage: F1 scales with model size; 3–4B models offer the strongest cost-quality balance. (b) Downstream stages: Claude-3.5-Sonnet and Claude-3-Opus lead; open-source models trail by 2–4 points.

We detail the convergence behavior of TaxCE’s iterative refinement loop (Section 4.6) across all three datasets. For our experiments, we set $\theta_E = 83$, $\theta_X = 75$, and $\theta_G = 74$ based on the following considerations: (i) $\theta_E = 83$ ensures leaf topics are sufficiently distinct for downstream classification without requiring near-perfect separation, which is impractical for domains with inherently related topics; (ii) $\theta_X = 75$ requires that at least three-quarters of corpus-derived concepts are covered, balancing coverage against the diminishing returns of capturing extremely rare intents; and (iii) $\theta_G = 74$ ensures leaf topics are specific enough to be actionable while accommodating domains where some concepts are inherently broad. Table 8 reports the iteration-wise EEG scores, threshold satisfaction status, and the corrective action applied at each step.

Across all datasets, the initial taxonomy (iteration 0) consistently fails to meet θ_E , triggering exclusivity-focused refinement first (reducing c to c'). For CFPB and AskUbuntu, this single correc-

Dataset	Iter.	$\mathcal{E}$ ($\theta_E=83$)	$\mathcal{X}$ ($\theta_X=75$)	$\mathcal{G}$ ($\theta_G=74$)	Action Taken
Flipkart	0	79.3 ✗	71.2 ✗	72.4 ✗	Low $\mathcal{E}$: re-group with $c' < c$, regenerate $\mathcal{K}^*$
	1	85.1 ✓	73.8 ✗	74.1 ✓	Low $\mathcal{X}$: re-group with $c''' > c$, relax standardization
	2	85.7 ✓	78.3 ✓	76.1 ✓	Converged
CFPB	0	82.4 ✗	74.8 ✗	75.1 ✓	Low $\mathcal{E}$: re-group with $c' < c$, regenerate $\mathcal{K}^*$
	1	87.1 ✓	80.5 ✓	78.2 ✓	Converged
AskUbuntu	0	80.1 ✗	70.4 ✗	71.8 ✗	Low $\mathcal{E}$: re-group with $c' < c$, regenerate $\mathcal{K}^*$
	1	86.3 ✓	76.9 ✓	75.6 ✓	Converged

Table 8: Iteration-wise convergence of EEG metrics using general default thresholds ($\theta_E=83$, $\theta_X=75$, $\theta_G=74$). ✓ meets threshold; ✗ below threshold. Actions follow the priority order $\mathcal{E} \rightarrow \mathcal{G} \rightarrow \mathcal{X}$ described in Section 4.6. Notation: $c' < c$ reduces groups (consolidates overlapping topics); $c''' > c$ increases groups (recovers missing coverage). Final converged scores match Table 1.

tion simultaneously brings all three metrics above their thresholds, as concept regeneration with fewer, larger groups naturally improves coverage and specificity. Flipkart requires a second iteration: after exclusivity is resolved, exhaustivity remains below θ_X , triggering a coverage-focused correction (increasing c to c''' and relaxing standardization aggressiveness) that recovers previously missed concepts. CFPB’s granularity already meets θ_G at iteration 0 ($75.1 > 74$), reflecting the structured nature of financial complaint language where concepts naturally map to specific topics.

In fewer than 5% of runs across all datasets, convergence is not achieved within 3 iterations due to high thematic ambiguity; the final taxonomy is returned with unmet thresholds flagged for manual review.

G Stage-wise evaluation

We evaluate the quality of intermediate outputs at three key stages of the TaxCE pipeline: (i) Key Information Extraction, (ii) Semantic Grouping, and (iii) Concept Generation and Standardization. This stage-wise analysis validates that each component produces high-quality outputs that propagate to the final taxonomy.

Key Information Extraction. We evaluate F_{extract} by measuring precision (P), recall (R), and F1 of actionable segment extraction against a manually labeled sample of 200 feedback texts per dataset (Figure 4(a)). CFPB achieves the highest F1 (90.0) due to its structured complaint language with clear issue descriptions. AskUbuntu scores lowest (83.7) because technical jargon and code snippets complicate segment boundary detection. Flipkart (85.8) reflects the additional challenge of multilingual extraction and translation. Across all datasets, precision exceeds recall, indicating that F_{extract} is conservative in extraction, preferring

to miss borderline segments rather than introduce noise.

Semantic Grouping. We evaluate grouping quality using three standard intrinsic metrics (Table 9): Silhouette Score (Rousseeuw, 1987), which measures how similar each point is to its own group versus neighboring groups (range $[-1, 1]$, higher is better); Calinski-Harabasz (CH) Index (Caliński and Harabasz, 1974), the ratio of between-group to within-group dispersion (higher indicates better-separated groups); and Davies-Bouldin (DB) Index (Davies and Bouldin, 1979), the average similarity between each group and its most similar group (lower indicates better separation). CFPB produces the best-separated groups across all three metrics, consistent with its structured language where complaint categories are well-delineated. AskUbuntu shows the weakest separation, reflecting the overlapping nature of technical support topics (e.g., networking issues spanning both hardware and software).

Concept Quality. We evaluate concept quality before and after standardization using an LLM-as-judge (Claude-Opus-4.6) that rates each concept on two criteria (3-point scale, Figure 4(b)): *coherence* (whether representative verbatims consistently describe the same concept) and *distinctness* (whether the concept is clearly different from other concepts). Standardization consistently improves distinctness (+0.7 avg across datasets) by merging redundant cross-group concepts, while coherence remains stable or slightly improves (+0.1 avg), confirming that merging does not conflate unrelated intents.

H Baseline setup and configuration

We describe the setup for each baseline evaluated in Section 6. To ensure fair comparison, all embedding-based methods use the same Sentence-

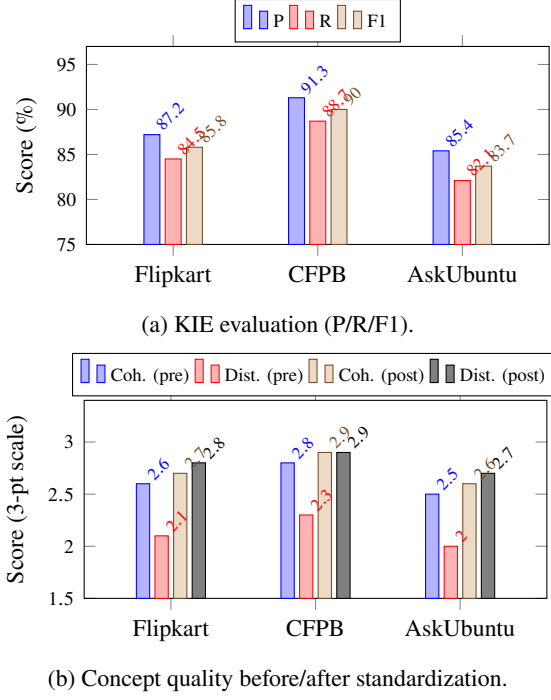


Figure 4: Stage-wise evaluation. (a) KIE achieves F1 83.7–90.0; CFPB’s structured language yields the highest scores, AskUbuntu’s technical jargon the lowest. (b) Standardization improves distinctness (+0.7 avg) while maintaining coherence.

Dataset	Silhouette↑	CH Index↑	DB Index↓
Flipkart	0.32	1842	1.45
CFPB	0.38	2534	1.21
AskUbuntu	0.31	1623	1.52

Table 9: Semantic grouping quality. ↑: higher is better; ↓: lower is better.

BERT model as TaxCE, and all LLM-based methods use the same downstream LLM. Flat methods (LDA, BERTopic, TopClus) produce unstructured topic sets evaluated as leaf-level topics; hierarchical methods (hLDA, LLM Zero-shot, LLM + Clustering) produce multi-level structures evaluated on the full hierarchy. As described in Section 6.1, exhaustivity and granularity for all methods are computed against the shared KIE-extracted reference S .

LDA. We use the Gensim⁹ implementation of Latent Dirichlet Allocation with standard preprocessing (tokenization, stopwords removal, lemmatization). The number of topics is set to 50, selected via C_v coherence score optimization (Röder et al., 2015) over the range {20, 30, 50, 75, 100}. Each topic is represented by its top-10 keywords concatenated as the topic label for EEG metric computation.

⁹<https://radimrehurek.com/gensim/>

We use default hyperparameters ($\alpha = 1/K$, $\eta = 1/K$ where K is the number of topics) with 1000 training iterations.

hLDA. We use the Tomotopy¹⁰ implementation of hierarchical LDA. The tree depth is set to 3 to match TaxCE’s default depth $h = 3$, enabling direct comparison of hierarchical structure quality. Documents are preprocessed identically to LDA. We use default hyperparameters ($\alpha = 0.1$, $\eta = 0.01$, $\gamma = 0.1$) with 1000 training iterations. Each topic at each level is represented by its top-10 keywords concatenated as the topic label.

BERTopic. We use the official BERTopic library¹¹. Documents are embedded using the same Sentence-BERT model as TaxCE to isolate the effect of the topic modeling approach from embedding quality. Dimensionality reduction uses UMAP ($n_neighbors=15$, $n_components=5$, $min_dist=0.0$) followed by HDBSCAN clustering ($min_cluster_size=15$). Topic representations are generated using class-based TF-IDF. We retain all discovered topics except the outlier topic (topic -1), which contains documents not assigned to any coherent cluster.

TopClus. We use the official implementation¹² of TopClus, which discovers topics via latent space clustering of pretrained language model representations. The number of topics is set to 50 to approximately match the number of leaf topics produced by TaxCE. All other hyperparameters follow the defaults specified in the original paper.

LLM Zero-shot. We prompt the same downstream LLM used in TaxCE’s pipeline stages (F_{CG} , F_{GT} , F_{tax}) to generate a 3-level taxonomy directly from a random sample of documents. The prompt instructs the LLM to read the provided documents, identify main themes and sub-themes, and organize them into a hierarchy with topic names and definitions. We use temperature 0.3 for near-deterministic generation. Due to context window limitations, the LLM cannot process the entire corpus; we experiment with sample sizes of 500, 1000, and 2000 documents and report the best-performing configuration. This baseline isolates the value of TaxCE’s progressive condensation pipeline over direct LLM prompting.

¹⁰<https://bab2min.github.io/tomotopy/>

¹¹<https://github.com/MaartenGr/BERTopic>

¹²<https://github.com/yumeng5/TopClus>

LLM + Clustering. This baseline combines embedding-based grouping with LLM-based topic naming, representing a simplified version of TaxCE without concept generation (F_{CG}) or standardization (F_{CS}). Documents are embedded using the same Sentence-BERT model and grouped using the same HDBSCAN configuration as TaxCE. The downstream LLM is then prompted with a sample of documents from each group to generate a descriptive topic name per group and organize the group topics into a 3-level hierarchy. This baseline isolates the contribution of TaxCE’s concept generation and standardization stages by keeping all other components identical.

Detailed performance analysis. Flat topic models (LDA, BERTopic, TopClus) achieve reasonable exclusivity (76–83) because their clustering mechanisms naturally separate topics, but score poorly on exhaustivity (28–47) and granularity (36–53). These methods identify only the most prominent topics, missing the long tail of specific intents. Their topics are represented as word distributions or cluster centroids rather than actionable definitions, limiting granularity. BERTopic outperforms LDA and TopClus due to its use of transformer embeddings, which capture richer semantic structure.

LLM Zero-shot achieves moderate exhaustivity (48–55) and granularity (53–57) by leveraging the LLM’s world knowledge to generate specific topics, but suffers from low exclusivity (66–70) because topics generated in a single pass without corpus-grounded deduplication frequently overlap. LLM + Clustering improves on this by grounding topic generation in corpus groups, raising exclusivity to 72–76 and exhaustivity to 55–60. However, feeding raw grouped feedback directly to the LLM without intermediate condensation leads to two key issues: (a) redundant information within groups overwhelms the LLM’s context window and dilutes its focus, causing it to either hallucinate topics not grounded in the corpus or miss specific intents buried in repetitive content, and (b) redundancies across groups persist since no corpus-wide deduplication is performed, producing overlapping leaf topics. TaxCE’s concept generation addresses (a) by condensing each group into atomic, non-redundant semantic units that provide the LLM with precisely the information needed for grounded topic generation; concept standardization addresses (b) by eliminating cross-group redundancy; and iterative refinement closes remaining metric gaps that neither

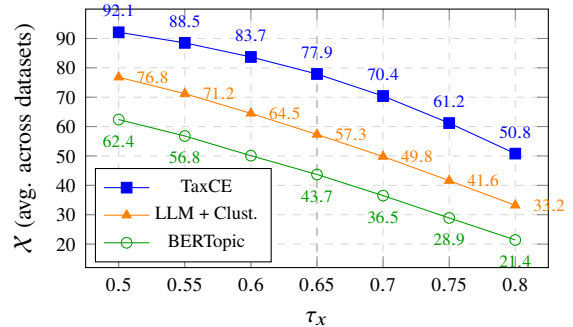


Figure 5: Exhaustivity (X) vs. coverage threshold τ_x , averaged across all three datasets. Method rankings are preserved across the full range.

Domain	Dataset	θ_E	θ_X	θ_G
Consumer Elec.	Flipkart	83	75	74
Financial Svc.	CFPB	84	77	76
Technical Sup.	AskUbuntu	83	75	74
General default		83	75	74

Table 10: EEG thresholds per dataset. Values represent minimum acceptable scores (0–100 scale).

clustering nor LLM generation alone can resolve.

I Exhaustivity threshold sensitivity

We vary the coverage threshold τ_x across [0.50, 0.80] and report exhaustivity averaged across all three datasets for TaxCE and the two strongest baselines (Figure 5). All embeddings use the same Sentence-BERT model for both segments and leaf topics across all methods.

All methods decline monotonically with increasing τ_x , as stricter thresholds demand closer segment-to-topic alignment. Method rankings remain consistent across the full range. We selected $\tau_x = 0.65$ by calibrating against manual coverage judgments on a held-out sample of 200 segments per dataset, where this value best separated genuine topical matches from spurious lexical overlap.

J EEG threshold guidelines

The quality thresholds θ_E , θ_X , θ_G depend on the domain characteristics, input structure, and intended downstream use of the taxonomy. Table 10 reports empirically determined EEG thresholds for each dataset, selected as the values that yield taxonomies rated highest on navigability and actionability by domain experts while remaining achievable within 1–3 refinement iterations (Appendix F).

CFPB supports the highest thresholds due to its structured complaint language with well-delineated

product categories. The general defaults ($\theta_E = 83$, $\theta_X = 75$, $\theta_G = 74$) serve as a practical starting point; practitioners should increase θ_E when the taxonomy feeds a classifier and increase θ_X for coverage-critical applications. For input types not evaluated here (e.g., chat transcripts, survey responses), we expect conversational inputs to require thresholds 2–3 points lower than structured inputs within the same domain due to increased ambiguity and topic overlap. Setting any threshold above 85 is generally impractical due to inherent semantic overlap in natural language; see Appendix L for a full stress-test quantifying graceful degradation and empirical ceilings.

K Embedding sensitivity

Because EEG relies on cosine similarity in embedding space, absolute scores depend on the embedding backbone. We re-ran the full evaluation on Flipkart under six embedding backbones spanning open-source Sentence-BERT variants, contrastive/retrieval models, and commercial APIs. Per-embedding coverage thresholds τ_x were recalibrated following Appendix I to control for scale differences across embedding spaces: S-BERT (Reimers and Gurevych, 2019) 0.65, MPNet (Song et al., 2020) 0.68, E5-large (Wang et al., 2024) 0.62, BGE-large (Xiao et al., 2024) 0.63, OpenAI-3-large (OpenAI, 2024) 0.58, OpenAI-3-small 0.61.

Embedding	$\mathcal{E}$	$\mathcal{X}$	$\mathcal{G}$
S-BERT (paper anchor)	85.7	78.3	76.1
MPNet-base-v2	83.2	75.8	73.4
E5-large-v2	87.4	79.6	77.8
BGE-large-en-v1.5	86.9	80.2	78.5
OpenAI-3-large	88.1	81.5	79.3
OpenAI-3-small	86.3	78.9	76.7

Table 11: TaxCE EEG scores on Flipkart across six embedding backbones with recalibrated τ_x . S-BERT is the deployment default used elsewhere in the paper and by all baselines (Appendix H); the other five backbones are evaluated here to quantify sensitivity.

Three findings hold across all six backbones: (a) the method ranking TaxCE > LLM+Clustering > BERTopic is preserved; (b) cross-embedding Pearson correlation of EEG scores is 0.89–0.97 for every pair, indicating that EEG measures a stable underlying property rather than an embedding-specific artefact; and (c) domain-tuned embeddings shift absolute scores but not rankings. For example, FinBERT (Araci, 2019) on CFPB pushes TaxCE

to 89.4/82.8/80.6, a +2.3 average gain over the general-purpose baseline. Refinement convergence also remains within the 1–3 iteration range for all six backbones on AskUbuntu, the most technically dense corpus. Absolute EEG scores are therefore recalibrateable via τ_x , while rankings and convergence behaviour are robust to embedding choice.

L Threshold stress-test

To quantify robustness of the refinement loop under aggressive thresholds, we shifted the defaults $(\theta_E, \theta_X, \theta_G) = (83, 75, 74)$ upward by +2, +3, +5, and +7 points simultaneously and capped iterations at 5.

Shift	Conv. rate	Avg. iter.	Fallback
default	$\geq 95\%$	1.3	$< 5\%$
+2	78%	2.4	22%
+3	42%	3.8	58%
+5	8%	4.9	92%
+7	0%	5.0	100%

Table 12: Refinement behaviour under threshold shifts. *Conv. rate*: fraction of runs meeting all three shifted thresholds within 5 iterations. *Fallback*: fraction returning the best-so-far taxonomy rather than a converged one.

Three empirical facts emerge. (a) Hard ceilings exist at $\mathcal{E} \approx 87$ –88, $\mathcal{X} \approx 82$ –83, $\mathcal{G} \approx 79$ –80 across all three datasets, reflecting genuine semantic overlap in natural language and validating the 85-ceiling claim in Appendix J. (b) When a target is unreachable, the best-so-far taxonomy is within 2–4% of default-threshold quality on all three metrics, so degradation is graceful rather than catastrophic. (c) The $\mathcal{E} \rightarrow \mathcal{G} \rightarrow \mathcal{X}$ priority order is empirically justified: pursuing $\mathcal{E}$ aggressively (re-grouping with smaller c to hit a raised θ_E) reduces coverage: $\mathcal{X}$ drops from 78.3 to 71.8 on Flipkart across three extra iterations, which is why the loop diagnoses and prioritises before acting rather than optimising each metric independently.

M EEG–human rubric correlation

To assess whether EEG tracks human perception of taxonomy quality, we compute Pearson and Spearman correlations between the three EEG metrics ($\mathcal{E}, \mathcal{X}, \mathcal{G}$) and the three human rubrics (Quality, Actionability, Navigability) used in the human evaluation (Section 6.4), across the three evaluated methods (BERTopic, LLM+Clustering, TaxCE).

$\mathcal{E}$ correlates strongly with all three rubrics, with the strongest tie to Navigability (Pearson $r \approx 0.72$) and Quality ($r \approx 0.78$), consistent with distinct sibling topics being easier to navigate and perceived as more coherent. $\mathcal{X}$ and $\mathcal{G}$ correlate moderately-to-strongly with Actionability ($r \approx 0.65$ – 0.71), with TaxCE consistently at the top of the range. Spearman correlations track Pearson values within 0.02–0.05 across all nine (metric, rubric) cells. In each cell, TaxCE’s per-method correlations are 0.10–0.25 points higher than baselines’, indicating that EEG tracks perceived quality more faithfully as taxonomy quality itself improves. Human rubrics were held out during refinement, so this alignment is not an artefact of metric-driven optimisation.

We position EEG as an intrinsic, corpus-grounded proxy that is predictive of, but does not replace, human judgement.

N Cost and runtime

We measured end-to-end cost on a 1M-document corpus using a single mid-range cloud GPU instance (24 GB, A10G-class) for the locally-hosted KIE tier and a commercial API for downstream stages. Total wall-clock time was under 10 hours, of which KIE accounted for approximately 80%; refinement averaged 2 iterations. KIE contributes no API cost. Of total API spend, concept generation and standardisation account for approximately 63% and hierarchy construction plus validation approximately 29%. Per-document cost decreases by roughly 25–30% as corpus size scales from 100K to 10M documents, since downstream costs are largely fixed with respect to corpus size. Substituting alternative frontier APIs changes total cost by –3% to +37% with under one EEG point of variation. The two-tier design is deliberate: locally-hosted 3–4B models absorb the high-volume document-level stage, while the stronger commercial API is invoked only approximately 140 times per taxonomy regardless of corpus size.

O Prompt templates

This section presents the generalized prompt templates used in each stage of the TaxCE pipeline. All templates use placeholders ($\{ \dots \}$) instantiated per dataset and use case.

O.1 Key information extraction (F_{extract})

F_{extract} : Key Information Extraction

You are an expert at extracting actionable insights from $\{\text{feedback_type}\}$ feedback related to $\{\text{domain_description}\}$.

Task: Read the feedback below and extract all explicitly stated actionable insights. Do not infer, guess, or extrapolate beyond what is directly stated. If the feedback is in a non-English language, translate the extracted segments and intents into $\{\text{target_language}\}$.

Rules:

1. Ignore empty, non-informative, or placeholder content.
2. Each insight must be directly stated by the user.
3. If no actionable insights exist, return only: `<output></output>`
4. Extract the following fields for each insight: $\{\text{field_definitions}\}$

Output format:

```
<output>
  <insight>
    <segment>exact verbatim text (translated if
non-English)</segment>
    <intent>semantic label summarizing the segment</intent>
    {additional_fields}
  </insight>
</output>
```

Do not output anything outside the `<output>` block.

`<feedback>\{\text{feedback\_text}\}</feedback>`

Placeholders: $\{\text{feedback_type}\}$: input type (e.g., review, transcript, survey); $\{\text{domain_description}\}$: domain context; $\{\text{target_language}\}$: taxonomy language (default: English); $\{\text{field_definitions}\}$: use-case fields (e.g., category, aspect, root cause, polarity); $\{\text{additional_fields}\}$: optional XML fields; $\{\text{feedback_text}\}$: raw input document.

O.2 Concept generation (F_{CG})

F_{CG} : Concept Generation

You are an expert feedback analyst. Given a group of semantically related feedback segments, identify all distinct, non-overlapping, and granular concepts discussed within the group.

Instructions:

- Extract concepts that are as specific and granular as possible.
- Ensure concepts are mutually exclusive within this group.
- For each concept, select up to $\{k\}$ representative segments that are diverse and collectively describe the concept’s full scope.
- Only include concepts supported by at least $\{k\}$ segments.
- Do not infer concepts not present in the input.
- Do not merge distinct concepts even if they are related.

Output format:

```
<output>
  <concept>
    <name>specific concept name</name>
    <representative_segments>
      <segment>verbatim segment 1</segment>
      <segment>verbatim segment 2</segment>
    </representative_segments>
  </concept>
</output>
```

Output only the `<output>` block.

`<group_segments>\{\text{segment\_list}\}</group_segments>`

Placeholders: $\{k\}$: minimum segment support and maximum representative segments per concept; $\{\text{segment_list}\}$: feedback segments from one semantic group.

O.3 Concept standardization (F_{CS})

F_{CS} : Concept Standardization

You are an expert at deduplicating and standardizing feedback concepts. Given a batch of concepts (each with representative segments), identify semantically equivalent concepts and merge them into canonical forms.

Instructions:

- Two concepts are equivalent if they describe the same underlying issue, even with different phrasing.
- Merge equivalent concepts into a single canonical concept with a standardized name.
- Consolidate representative segments from merged concepts, removing semantic duplicates while preserving diversity.
- Do not merge concepts that are related but distinct.
- Preserve the specificity and granularity of each concept.

Output format:

```
<output>
  <standardized_concept>
    <name>canonical concept name</name>
    <merged_from>
      <original>original concept name 1</original>
    </merged_from>
    <representative_segments>
      <segment>segment 1</segment>
    </representative_segments>
  </standardized_concept>
</output>
```

Output only the <output> block.

```
<concept_batch>{concept_batch}</concept_batch>
```

Placeholders: {concept_batch}: batch of semantically related concepts with representative segments (JSON or XML).

O.4 Granular topic and definition generation (F_{GT})

F_{GT} : Granular Topic & Definition Generation

You are an expert feedback analyst. Given standardized concepts with representative segments, generate granular topics grouped under latent parent topics.

Instructions:

- Merge overlapping concepts into exclusive, non-overlapping granular topics.
- Group related granular topics under latent parent topics (abstract, neutral names). These parent topics are used only for lightweight clustering and are not included in the final taxonomy.
- Each parent topic must contain at least {min_children} granular topics.
- For each granular topic, provide:
 - A specific, polarity-labeled name (max {max_words} words)
 - Polarity: positive or negative
 - Definition: 2-3 sentence description of the topic scope
 - Examples: at least {min_examples} diverse representative segments, unique to this topic (no overlap with sibling topics)
- Generate a catch-all topic per parent for generic feedback lacking specificity, with a descriptive name (do not use "catch all").
- All names: lowercase, space-separated, alphanumeric only.
- Do not infer topics not supported by the input.

Output format:

```
<topics>
  <parent_topic>
    <name>latent parent topic name</name>
    <granular_topics>
      <granular_topic>
        <name>granular topic name</name>
        <polarity>positive/negative</polarity>
        <definition>2-3 sentence definition</definition>
        <examples>example1; example2; ...</examples>
      </granular_topic>
    </granular_topics>
  </parent_topic>
</topics>
```

Output only the <topics> block.

```
<standardized_concepts>{concept_dict}</standardized_concepts>
```

Placeholders: {max_words}: max words per topic name; {min_children}: min granular topics per parent; {min_examples}: min examples per topic; {concept_dict}: standardized concepts with representative segments.

O.5 Hierarchical taxonomy construction (F_{tax})

F_{tax} : Hierarchical Taxonomy Construction

You are an expert at organizing topics into hierarchical taxonomies. Given a set of hinge topics (L2) and their granular topics (L3), construct the top level (L1) by grouping semantically related hinge topics under abstract coarse category names.

Instructions:

- Each L1 category must group at least {min_l2} hinge topics sharing a common theme.
- L1 names must be abstract, neutral, and holistic.
- Preserve the existing L2-L3 structure unchanged.
- Ensure L1 categories are mutually exclusive and collectively exhaustive of all hinge topics.
- All names: lowercase, space-separated, alphanumeric only.
- Do not create, modify, or remove any L2 or L3 topics.

Output format:

```
<taxonomy>
  <l1_category>
    <name>coarse category name</name>
    <l2_topics>
      <l2_topic>existing hinge topic name</l2_topic>
    </l2_topics>
  </l1_category>
</taxonomy>
```

Output only the <taxonomy> block.

```
<hinge_topics>{hinge_topic_list}</hinge_topics>
```

Placeholders: {min_l2}: min hinge topics per L1 category; {hinge_topic_list}: all L2 hinge topic names from F_{GT} . For $h > 3$, F_{tax} is applied recursively at each level.

O.6 Root-to-leaf validation (F_{val})

F_{val} : Root-to-Leaf Validation

You are an expert at evaluating hierarchical taxonomy quality. Given a list of root-to-leaf paths from a taxonomy and a sample of corpus feedback, validate each path.

For each path (L1 → L2 → L3), verify:

1. Logical coherence: the progression from coarse to hinge to granular represents a semantically meaningful specialization.
2. Corpus support: at least one feedback segment in the provided corpus sample supports the topic combination along this path.

Output format:

```
<validation>
  <path>
    <l1>coarse topic</l1>
    <l2>hinge topic</l2>
    <l3>granular topic</l3>
    <coherent>yes/no</coherent>
    <supported>yes/no</supported>
    <reason>brief explanation if no</reason>
  </path>
</validation>
```

Output only the <validation> block.

```
<paths>{path_list}</paths>
<corpus_sample>{corpus_sample}</corpus_sample>
```

Placeholders: {path_list}: all root-to-leaf paths from the constructed taxonomy; {corpus_sample}:

representative feedback texts from D for grounding verification.

P Phase-wise examples

This section illustrates each stage of the TaxCE pipeline with concrete examples from the Flipkart dataset, emphasizing non-obvious cases that highlight the framework’s contributions. All verbatims are drawn from actual corpus content.

Granularity vs. exhaustivity illustration. To clarify the distinction between exhaustivity and granularity: given concepts “*fast battery drain*”, “*broken noise cancellation*”, and “*loose charging connection*”, a single leaf topic “*Battery & Power Systems*” yields high exhaustivity (all three segments are loosely covered) but low granularity (the topic is too broad to closely match any individual segment). In contrast, three matching specific topics yield both high exhaustivity and high granularity, as each segment maps tightly to its corresponding leaf.

P.1 Key Information Extraction (F_{extract})

KIE extracts atomic actionable segments and assigns intent labels jointly, enriching each segment with document-level context. We illustrate with two examples: a standard case and a case where intent enrichment resolves segment-level ambiguity.

Example 1: Multi-polarity Document

“I bought this tablet for my daughter. The screen is very bright and clear, but the battery dies within 3 hours of use. Also, the charging cable that came with it stopped working after a week.”

Extracted Intent-Tuples

Segment (s)	Intent (I)
“The screen is very bright and clear”	Positive display quality
“the battery dies within 3 hours of use”	Fast battery drain
“the charging cable stopped working after a week”	Defective charging cable

A single document yields three intent-tuples spanning different L1 categories (display, battery, accessories) and polarities (positive, negative, negative), demonstrating TaxCE’s ability to decompose multi-topic documents into atomic units.

Example 2: Ambiguous Segment Requiring Intent Enrichment

“This Smart doorbell is great but it keeps going offline every few hours. I called support and they told me to move my router closer, but that didn’t help at all. At least the night vision is crystal clear.”

Grp	Sample Segments (s)	Sample Intents (I)
C_{14}	“battery dies within 3 hours of use”	Fast battery drain
	“battery life is insane, lasts forever”	Excellent battery duration
	“battery % jumps from 40 to dead”	Unreliable battery indicator
	“batteries died after only 3 months”	Early battery failure
C_{27}	“will not charge and is completely dead”	Total charging failure
	“charges super fast with USB”	Rapid charging performance
	“charging for 14 hours only 65%”	Extremely slow charging rate
	“charging port melted the plastic”	Dangerous port melting
C_{41}	“cameras are offline more than online”	Frequent camera disconnections
	“WiFi signal does not reach area”	Insufficient WiFi signal range
	“connected right away without issues”	Easy WiFi connection

Table 13: Semantic grouping of intent-tuples. Groups are semantic neighborhoods containing multiple latent concepts that concept generation (Section P.3) will subsequently separate.

Extracted Intent-Tuples

Segment (s)	Intent (I)
“it keeps going offline every few hours”	Frequent camera disconnections from WiFi
“they told me to move my router closer, but that didn’t help”	Ineffective troubleshooting by support
“the night vision is crystal clear”	Excellent nighttime video clarity

The segment “they told me to move my router closer, but that didn’t help” is ambiguous in isolation: it could describe a router range issue or a support quality issue. The intent label *Ineffective troubleshooting by support* disambiguates by incorporating document context (the user sought help and received unhelpful advice), directing this tuple toward customer support topics rather than WiFi range topics during grouping.

P.2 Semantic Grouping

Intent-tuples from across the corpus are embedded and grouped into c semantically coherent clusters. The grouping creates *semantic neighborhoods*, not final topics. Table 13 illustrates three groups formed from battery, power, and connectivity-related tuples.

A few things are worth noting about these groupings. Group C_{14} ends up mixing positive and negative tuples: “battery life is insane” sits alongside “battery dies within 3 hours” because both are fundamentally about battery life, even though one is praise and the other a complaint. Polarity-based separation is not the clustering algorithm’s job; that falls to concept generation in the next stage. A more useful pattern emerges across groups: “battery dies within 3 hours” (C_{14}) and “will not charge and is completely dead” (C_{27}) are both power failures, but HDBSCAN places them in different clusters because C_{14} is organized around discharge behavior

while C_{27} is organized around the charging system. This is a useful distinction that the downstream stages can exploit, even though a human annotator might initially group them together.

P.3 Concept Generation (F_{CG})

For each group, F_{CG} identifies distinct concepts and selects representative verbatims. The critical contribution is separating multiple fine-grained intents that co-exist within a single semantic neighborhood. We illustrate with group C_{14} .

Input: Group C_{14} (Battery Life Neighborhood, 847 intent-tuples)

“battery dies within 3 hours”, “battery drains very quickly when unplugged”, “battery life is insane”, “outstanding battery life”, “battery percentage jumps around erratically”, “battery level is not accurate”, “batteries died after only 3 months”, “battery lasted only two months”, “need to charge twice a day”, “battery drops from 100% to 3% within hours”, ...

Output: Concepts $\mathcal{K}_{14}$ (4 concepts identified)

Concept 1: fast battery drain

Representative verbatims: “battery dies within 3 hours”; “battery drains very quickly when unplugged”; “battery drops from 100% to 3% within hours”; “need to charge twice a day”

Concept 2: excellent battery duration

Representative verbatims: “battery life is insane”; “outstanding battery life”; “battery life is phenomenal”

Concept 3: unreliable battery indicator

Representative verbatims: “battery percentage jumps around erratically”; “battery level is not accurate”; “battery percentage kept jumping”

Concept 4: early battery failure

Representative verbatims: “batteries died after only 3 months”; “battery lasted only two months”; “battery died prematurely”

To see why concept generation matters, consider the alternative: passing all 847 tuples from this group straight to the LLM. In our early experiments, this reliably produced one or two broad topics along the lines of “battery issues” or “battery life problems.” The LLM simply could not attend to the finer distinctions buried across hundreds of repetitive complaints. With concept generation, we get four distinct outputs. The most important split is between fast battery drain and early battery failure. Both involve batteries that “die,” and customers often use similar language for both, but they point to very different problems. The first is about a battery that drains too quickly on a given day; the second is about a battery that stops holding charge altogether after a few months. They need different teams to investigate and different resolutions, so conflating them would undermine the taxonomy’s usefulness.

P.4 Concept Standardization (F_{CS})

Since grouping is inherently imperfect, semantically same concepts may emerge independently in different groups. Standardization merges these

while preserving genuinely distinct concepts. We illustrate both a **merge** case and a critical **non-merge** case.

Merge Case: Semantically Same Concepts from Different Groups

From Group C_{14} : fast battery drain

“battery dies within 3 hours”; “battery drains very quickly when unplugged”

From Group C_{31} : rapid battery depletion

“phone dies in 2 hours”; “battery runs out quickly”; “drains batteries within 2 days”

Merge Output: Standardized Concept κ_j^*

Standardized concept: fast battery drain

Merged from: *fast battery drain* (C_{14}) + *rapid battery depletion* (C_{31})

Consolidated verbatims: “battery dies within 3 hours”; “phone dies in 2 hours”; “battery drains very quickly when unplugged”; “battery runs out quickly”

(Removed: “drains batteries within 2 days” as semantically overlapping with “battery drains very quickly when unplugged”)

The two concepts describe the same phenomenon (abnormally rapid charge depletion during normal use) with different surface forms. Note that verbatim consolidation removes semantic overlaps while preserving diversity.

Non-Merge Case: Related but Distinct Concepts

Candidate 1: fast battery drain

“battery dies within 3 hours”; “battery drains very quickly when unplugged”

Candidate 2: early battery failure

“batteries died after only 3 months”; “battery lasted only two months”

Decision: Not merged. *Fast battery drain* describes rapid depletion within a single charge cycle (hours), while *early battery failure* describes permanent hardware degradation over the product lifespan (months). They require different resolutions (software optimization vs. warranty replacement) and map to different actionable intents.

Getting this non-merge decision right matters a lot in practice. If standardization is too aggressive here, both concepts collapse into something like “battery problems,” and the distinction between a user who needs to tweak their power settings and one who should file a warranty claim is lost entirely. That is exactly the kind of granularity a taxonomy needs to preserve to be useful downstream. The moderate setting we describe in Section 4.3 is designed to catch genuine duplicates (like the “fast battery drain” / “rapid battery depletion” pair above) while leaving cases like this one alone, where the surface-level similarity masks a real difference in what went wrong.

P.5 Granular Topic and Definition Generation (F_{GT})

F_{GT} transforms standardized concepts into leaf topics with natural language definitions, polarity labels, and illustrative examples. Concepts that are directly suitable become individual topics; closely overlapping concepts merge. A catch-all topic is

generated per coarse category.

Input: Standardized Concepts (battery life neighborhood)

κ_1^* : fast battery drain κ_2^* : excellent battery duration κ_3^* : unreliable battery indicator
 κ_4^* : early battery failure κ_5^* : excessive battery replacement frequency κ_6^* : battery drains fast on remote control

Output: Granular Topics Under Latent Parent “battery life and performance”

Topic g_1 : fast battery drain [negative]

Definition: Describes rapid and unexpected depletion of battery charge during normal device usage, where the battery power decreases significantly within 24–48 hours of charging. This issue manifests as unusually quick power loss that significantly impacts the device’s usability and requires frequent recharging.

Examples: “battery dies within 3 hours”; “battery drops from 100% to 3% within hours”; “battery completely drained within a day”

Topic g_2 : excellent battery duration [positive]

Definition: Refers to the exceptional and impressive duration of battery life during normal device usage. The battery demonstrates outstanding longevity, allowing for extended use between charges that exceeds typical expectations.

Examples: “battery life is insane”; “battery lasts literally forever”; “outstanding battery life”

Topic g_3 : unreliable battery indicator [negative]

Definition: Describes issues with the device’s battery level monitoring system displaying erratic or inconsistent readings. The battery percentage indicator shows incorrect values or jumps between different levels unexpectedly, making it impossible to accurately determine remaining battery life.

Examples: “battery percentage jumps around erratically”; “battery level is not accurate”

Topic g_4 : early battery failure [negative]

Definition: Indicates complete battery failure or significant degradation occurring within the first few months of device usage, well before the expected lifespan. This represents a critical hardware issue where the battery becomes unusable much earlier than its intended service life.

Examples: “batteries died after only 3 months”; “battery lasted only two months”

Topic g_5 : rapid battery depletion by remote [negative]

Definition: Refers to the extremely short battery life and unusually quick power consumption of the remote control specifically. The batteries drain at an abnormally fast rate, requiring frequent replacement within days or even hours of normal usage.

Examples: “remote eats batteries like nothing”; “batteries lasting only about two days”

Several decisions warrant explanation. First, concepts κ_1^* (fast battery drain) and κ_6^* (battery drains fast on remote control) are not merged despite surface similarity: κ_1^* describes tablet/camera battery issues while κ_6^* is remote-specific, requiring different product teams to action. Second, κ_5^* (excessive battery replacement frequency) is merged into g_4 (early battery failure) because the replacement frequency is a consequence of premature failure rather than a distinct actionable topic. Third, definitions are written to be usable by downstream classifiers: they specify scope boundaries (“within 24–48 hours” for drain vs. “within the first few months” for failure) that help distinguish ambiguous feedback.

P.6 Hierarchical Taxonomy Construction

(F_{tax})

The granular topics $\mathcal{G}$ form the leaf level L_3 . F_{tax} constructs the hierarchy bottom-up: first grouping L_3 topics into L_2 hinge topics, then grouping L_2 topics into L_1 coarse categories. We trace this process for a subset of the “battery and power systems” branch.

Step 1: $L_3 \rightarrow L_2$ Grouping (Granular $\rightarrow$ Hinge)

F_{tax} identifies that the following L_3 topics share a common theme of battery operational behavior:

g_1 : fast battery drain g_2 : excellent battery duration g_3 : unreliable battery indicator
 g_4 : early battery failure g_5 : rapid battery depletion by remote ... (8 more)

$\Rightarrow$ **Generated L_2 :** battery life and performance

Separately, these L_3 topics share a common theme of charging system functionality:

g_{18} : total charging failure g_{19} : rapid charging performance g_{20} : extremely slow charging rate
 g_{21} : charging system breakdown g_{22} : persistent charging failure ... (15 more)

$\Rightarrow$ **Generated L_2 :** charging problems

And these L_3 topics share a common theme of device power supply and management:

g_{38} : complete power failure g_{39} : efficient solar power generation
 g_{40} : unstable power operation
 g_{41} : missing wired power option ... (20 more)

$\Rightarrow$ **Generated L_2 :** power management issues

Step 2: $L_2 \rightarrow L_1$ Grouping (Hinge $\rightarrow$ Coarse)

F_{tax} identifies that the following L_2 hinge topics describe related aspects of power and energy systems:

battery life and performance charging problems power management issues
 battery hardware defects battery design and accessibility power adapter issues

$\Rightarrow$ **Generated L_1 :** battery and power systems

Resulting Hierarchy Branch (3 levels)

L1: battery and power systems

L2: battery life and performance

L_3 : fast battery drain [−] | excellent battery duration [+] | unreliable battery indicator [−] | early battery failure [−] | rapid battery depletion by remote [−] | ...

L2: charging problems

L_3 : total charging failure [−] | rapid charging performance [+] | extremely slow charging rate [−] | ...

L2: power management issues

L_3 : complete power failure [−] | efficient solar power generation [+] | unstable power operation [−] | ...

L2: power adapter issues

L_3 : broken power adapter [−] | loose power cord connection [−] | incorrect voltage rating [−] | ...

The bottom-up construction ensures every leaf is corpus-grounded (traceable through $\mathcal{K}^*$ to segments in D), while the generated L_2 and L_1 topics provide navigable abstractions. Root-to-leaf validation (F_{val} , Section 4.5) then verifies each path; for instance, the path *battery and power systems* $\rightarrow$ *charging problems* $\rightarrow$ *rapid charging performance* is validated as logically coherent (a positive charging speed observation correctly belongs under charging functionality within the power domain) and corpus-supported.