Hierarchical Text Classification with LLM-Refined Taxonomies

Jonas Golde[1,*]   Nicolaas Jedema[2]   Ravi Krishnan[3,*]   Phong Le[4,*]
[1]Humboldt Universität zu Berlin   [2]Amazon   [3]Meta
[4]School of Computer Science, University of St Andrews

Abstract

Hierarchical text classification (HTC) depends on taxonomies that organize labels into structured hierarchies. However, many real-world taxonomies introduce ambiguities, such as identical leaf names under similar parent nodes, which prevent language models (LMs) from learning clear decision boundaries. In this paper, we present TAXMORPH, a framework that uses large language models (LLMs) to transform entire taxonomies through operations such as renaming, merging, splitting, and reordering. Unlike prior work, our method revises the full hierarchy to better match the semantics encoded by LMs. Experiments across three HTC benchmarks show that LLM-refined taxonomies consistently outperform human-curated ones in various settings up to +2.9pp. in F1.

To better understand these improvements, we compare how well LMs can assign leaf nodes to parent nodes and vice versa across human-curated and LLM-refined taxonomies. We find that human-curated taxonomies lead to more easily separable clusters in embedding space. However, the LLM-refined taxonomies align more closely with the model's actual confusion patterns during classification. In other words, even though they are harder to separate, they better reflect the model's inductive biases. These findings suggest that LLM-guided refinement creates taxonomies that are more compatible with how models learn, improving HTC performance.

1 Introduction

Hierarchies and taxonomies are central to human cognition and language understanding. Organizing information into nested categories enables efficient storage, retrieval, and reasoning about complex concepts (Dhillon et al., 2002; Sun and Lim, 2001). In natural language processing, hierarchical
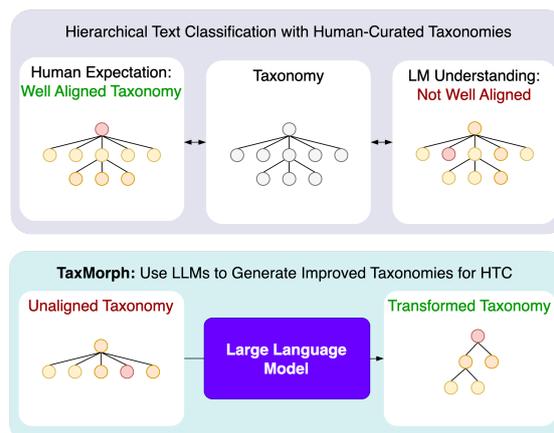

Figure 1: Human-curated taxonomies used in hierarchical text classification (HTC) are often suboptimal due to inconsistencies or ambiguities. TAXMORPH transforms these taxonomies into structures that better align with the internal representations of language models, leading to improved performance in HTC tasks.

structures manifest in taxonomies, ontologies, and knowledge bases (Vrandečić and Krötzsch, 2014; Lehmann et al., 2015; Heist and Paulheim, 2021).

Hierarchical text classification (HTC) requires models to assign documents to labels organized in a hierarchy. Recent studies have shown that large language models (LLMs) acquire rich hierarchical knowledge during pretraining (Lin and Ng, 2022; Wu et al., 2023). However, their effectiveness in HTC depends on how well their internal representations align with human-curated taxonomies. These taxonomies are manually constructed, often contain redundancies or ambiguities, and may not reflect how LLMs naturally organize or differentiate categories.

For instance, taxonomies may include duplicated or overly generic nodes (e.g., "Design" appearing under both "Web" and "Fashion") or conflicting class boundaries, which can prevent models from forming clear decision boundaries. Misalignment between the model's representations and the struc-
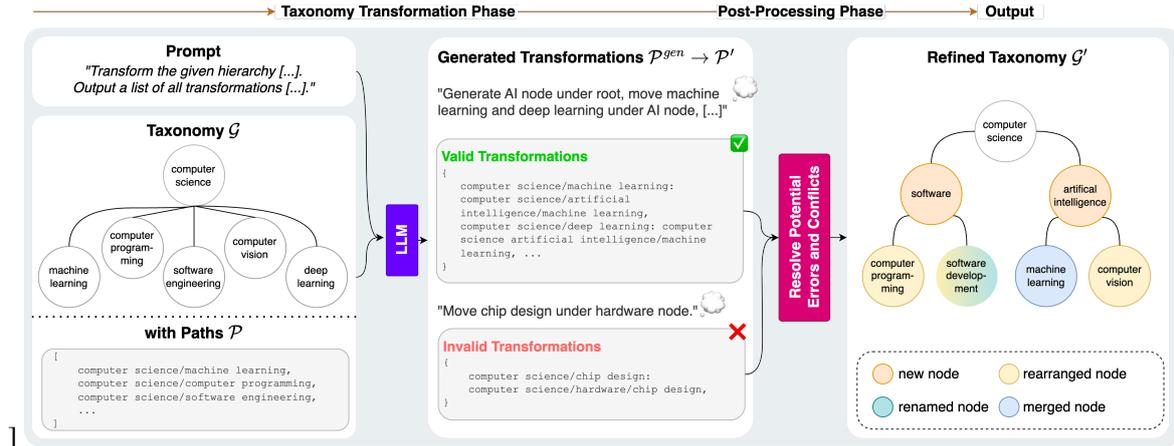

Figure 2: Overview of TaxMorph, a two-step framework for refining taxonomies. In the first phase, an LLM generates transformations by considering the full context of the input taxonomy $\mathcal{G}$. In the second phase, we apply post-processing to correct hallucinations and resolve inconsistencies, resulting in the final refined taxonomy $\mathcal{G}'$.

ture of human-defined hierarchies can introduce classification errors and limit performance in HTC.

In this paper, we study whether the structure of a given taxonomy is actually learnable by a language model, and how its structure can be improved to make categories more separable. We introduce TaxMorph, a two-step framework that transforms an existing taxonomy by renaming, merging, splitting, or reordering nodes using an LLM. Unlike prior work, our method operates over the full structure, effectively treating LLMs as taxonomists that revise entire hierarchies in a context-aware manner.

We evaluate our approach across three HTC datasets and show that LLM-refined taxonomies consistently improve classification performance over their human-curated counterparts. To better understand these gains, we analyze the separability of refined hierarchies in embedding space and find that LLM-generated structures better reflect model confusion patterns. These findings highlight the importance of taxonomy structure and suggest that LLMs can play a key role in improving them.

We summarize our contributions as follows:

- We propose TaxMorph, a framework for transforming taxonomies using LLMs to improve hierarchical text classification.

- We show that modeling with LLM-refined taxonomies consistently outperforms human-curated baselines across multiple datasets.

- We conduct detailed ablations to explain why refined hierarchies are more effective, showing they produce harder but more meaningful distinctions in embedding space.

2   TaxMorph

Creating effective taxonomies involves subtle decisions about granularity, naming, and consistency. These decisions are typically guided by domain knowledge and involve trade-offs similar to those in defining annotation schemes for tasks like named entity recognition (NER) (Wang et al., 2019; Rücker and Akbik, 2023). However, taxonomies introduce additional challenges, as they require structuring an entire concept space rather than classifying isolated entities (Kozareva and Hovy, 2010; Bansal et al., 2014). This often leads to ambiguous or inconsistent label structures, making it difficult for models to learn separable categories (He et al., 2024).

We propose TaxMorph, a framework that utilizes LLMs as taxonomists by letting the LLM transform and refine a given hierarchy into a more coherent and semantically meaningful structure. TaxMorph consists of two stages: (1) a generation phase that produces transformations across the full hierarchy and (2) a post-processing phase that resolves inconsistencies in the generated output.

2.1   Problem Setup

We focus on hierarchical text classification (HTC), where each input $\mathcal{X}_i \in \mathcal{X}$ is annotated with a subset $\mathcal{Y}_i \subset \mathcal{Y}$ of labels structured in a taxonomy $\mathcal{G} = (V, E)$. Here, V denotes the set of nodes (labels), and E the set of directed edges representing parent-child relationships. We assume $\mathcal{G}$ is a directed acyclic graph (DAG).

Each taxonomy can also be represented as a set of root-to-leaf paths. For each node $v \in V$, we

define a unique path $P_v = (v_r, \ldots, v)$ from some root node v_r to v, where each consecutive pair of nodes is connected by an edge in E. The entire taxonomy $\mathcal{G}$ can therefore be expressed as the set of all such paths: $\mathcal{P} = \{P_v \mid v \in V\}$.

2.2 Stage 1: LLM-Guided Taxonomy Transformation

In the first phase (cf. Figure 2, left), we use an LLM to generate a set of transformations over the original taxonomy $\mathcal{G}$. Concretely, we define a function f that maps each original path $P_v \in \mathcal{P}$ to a JSON object of the form {original_path: new_path}. To encourage holistic refinements, we prompt the LLM to process the entire taxonomy at once, rather than transforming each node individually. This full-context generation allows the model to propose more coherent restructurings, such as consistent renaming or merging across branches. However, generating the hierarchy in this way also requires keeping track of the original paths (the keys in the JSON object, denoted as $\mathcal{P}^{\text{gen}}$), as they serve as anchors for aligning the transformed paths $\mathcal{P}'$ with the original taxonomy. By comparing each original path with its transformed counterpart, we can identify the changes applied by the LLM. We categorize these transformations into four types:

- **Renamed:** A node's label is semantically adjusted to improve clarity or specificity.

- **Rearranged:** A node is reassigned to a different parent, reflecting a revised understanding of conceptual structure.

- **Generated:** New intermediate nodes are inserted into a path to enrich context or disambiguate meaning. We do not generate new leaf nodes.

- **Merged:** Multiple nodes with overlapping semantics are merged into a single representative node.

2.3 Stage 2: Post-Processing and Error Correction

In the second stage (cf. Figure 2, right), we post-process the JSON mapping generated by the LLM, where each key-value pair represents a transformation from an original path P_v^{gen} to a refined path P_v'. Ideally, each key P_v^{gen} corresponds exactly to an existing path P_v in the original taxonomy $\mathcal{P}$. However, LLM outputs may contain hallucinations

or minor deviations such as typos, incorrect formatting, or entirely fabricated entries.

To resolve such cases, we perform a matching step between P_v^{gen} and the true paths in $\mathcal{P}$ using normalized Levenshtein distance (Yujian and Bo, 2007). If a close match exists, a human reviewer verifies and corrects the key. Otherwise, we discard the transformation. This process allows us to preserve valid transformations while filtering out unreliable ones, all without modifying the original training data or requiring reannotation.

The final refined hierarchy $\mathcal{G}'$ is then constructed from the filtered and corrected set of path mappings $\mathcal{P}'$ which we then subsequently use in our experiments.

3 Experimental Setup

Datasets. We evaluate our method on four commonly-used hierarchical text classification benchmarks:

- **Amazon**[1]: Product reviews categorized under a hierarchy of product categories.

- **Books** (Aly et al., 2019b): English-language book blurbs categorized into genres using a four-level taxonomy.

- **Web of Science (WOS)** (Kowsari et al., 2017): Scientific paper abstracts organized by academic domain and subdomain.

Table 1 provides summary statistics, and Table 2 shows the number of labels at each depth level.

Dataset	Classes	Depth	Train	Val	Test
Amazon	584	3	37.7k	4.7k	4.7k
Books	152	4	58.7k	14.8k	18.4k
WOS	152	2	32.9k	7.0k	7.0k

Table 1: Dataset statistics, including total number of labels, maximum taxonomy depth, and split sizes.

Level	Amazon	Books	WOS
1	6	7	7
2	64	52	145
3	514	77	–
4	–	16	–

Table 2: Number of labels per level in each dataset's taxonomy.

---

[1] https://www.kaggle.com/datasets/kashnitsky/hierarchical-text-classification

3.1 Experimental Setup

Taxonomy Refinement with LLMs. To refine the taxonomies, we consider three LLMs: `Haiku`, `Sonnet-3`, and `Sonnet-3.5`. All models used have long-context capabilities and are sufficiently large to enable strong zero-shot performance, which is essential given the lack of specialized models for taxonomy refinement. For each model, we generate three independent refinements, following the procedure described in Section 2. This allows us to assess both model-dependent and stochastic variability in the refined hierarchies.

Hierarchical Classification Model. For downstream evaluation, we use a bi-encoder model that encodes both input texts $\mathcal{X}$ and taxonomy labels $\mathcal{Y}$ using shared parameters θ. Given a text input $\mathcal{X}_i$ and a candidate label $y \in \mathcal{Y}$, we compute their similarity using a dot product:

$$s = \theta(\mathcal{X}_i)^\top \cdot \theta(y),$$

where $\theta(\cdot) \in \mathbb{R}^d$. The resulting scores are passed through a sigmoid function to obtain probabilities. We train the model using binary cross-entropy loss, and during inference, we predict a label if its probability exceeds a threshold of 0.5.

We use DistilBERT (Sanh et al., 2020) as the encoder across all experiments. We employ AdamW optimizer (Loshchilov and Hutter, 2019), with a learning rate of 2×10^{-5}, a batch size 32, and a maximum of 10,000 training steps. All experiments are implemented using Hugging Face Transformers (Wolf et al., 2020) and PyTorch (Ansel et al., 2024).

Label Representations. We explore four label representations to assess the role of semantic and structural information in hierarchical classification.

- **Single Node:** Each label is represented using only its leaf node name (e.g., `machine learning`), without parent or ancestor context. This setting uses only local semantics.

- **Full Path:** Each label is represented by its full hierarchical path from the root (e.g., `computer science → artificial intelligence → machine learning`), providing additional contextual information that can help disambiguate similar leaf nodes.

To better isolate the contribution of label semantics, we also include two control baselines that remove natural language descriptions from the labels.

- **Linear Layer:** A standard classifier that treats labels as output indices with no associated text or structure. This setup tests whether the model can learn class boundaries without any semantic guidance.

- **Structured Identifier:** Each label is replaced by a code (e.g., `1.10.20`) that reflects its location in the hierarchy but omits all semantic content. This setting retains hierarchical structure while discarding label meaning.

4 Experiments

4.1 Experiment 1: Transformations Applied through the LLM

We analyze the transformations applied by TAX-MORPH across datasets and LLMs (cf. Table 3). Renaming is the most common change, particularly in deeper or more complex taxonomies. For example, Sonnet-3 renames 397 nodes in the Amazon taxonomy covering over two-thirds of its 584 labels. In WOS, Sonnet-3.5 reassigns 138 nodes to different parents, nearly matching the 145 total labels at level 2, indicating that the LLM reorganizes the majority of the hierarchy. We also observe a substantial number of generated nodes; for instance, Sonnet-3.5 inserts 82 new intermediate nodes in WOS to improve semantic separation between paths.

We further assess the reliability of these refinements by tracking invalid transformations, e.g. cases where the LLM generates paths not present in the original taxonomy. Such errors are rare and mostly confined to Haiku, which produces an average of 21 invalid paths in WOS. In contrast, Sonnet-3.5 yields zero invalid paths across all datasets, highlighting that more advanced LLMs can restructure large, real-world hierarchies with high precision. Overall, the scale and type of transformations correlate with the depth and complexity of the original taxonomy, suggesting that TAXMORPH effectively adapts to the structure of its input.

Finally, a human reviewer was consulted in only 0.6% of all cases, all of which involved minor typos in the LLM-generated paths. In each instance, our string matching procedure correctly identified the intended existing path, demonstrating its effectiveness in resolving such errors automatically.

Model	Dataset	Invalid	Renamed	Rearranged	Generated	Merged	Unchanged
Haiku	WOS	$20.7_{18.1}$	-	$118.7_{14.6}$	$26.3_{8.7}$	-	$17.0_{0.0}$
	Books	$10.0_{9.0}$	$78.0_{22.6}$	$6.5_{1.5}$	$2.5_{0.5}$	-	$31.0_{14.9}$
	Amazon	$1.5_{0.5}$	$257.0_{0.0}$	$38.0_{20.0}$	$12.5_{8.5}$	$14.0_{4.0}$	$52.0_{0.0}$
Sonnet (3)	WOS	-	$90.0_{4.0}$	$102.5_{2.5}$	$55.5_{2.5}$	-	$18.5_{1.5}$
	Books	-	$108.0_{7.0}$	-	-	-	$13.5_{6.5}$
	Amazon	$1.0_{0.0}$	$397.0_{3.0}$	-	-	$7.0_{0.0}$	$4.0_{0.0}$
Sonnet (3.5)	WOS	-	$126.0_{1.0}$	$138.0_{1.0}$	$82.0_{0.0}$	-	$1.0_{0.0}$
	Books	-	$108.0_{0.0}$	$2.5_{0.5}$	$2.0_{0.0}$	-	$8.5_{0.5}$
	Amazon	-	$36.2_{19.6}$	$1.0_{0.0}$	$1.0_{0.0}$	-	$2.2_{1.6}$

Table 3: Transformations applied through the LLM to the human-curated taxonomy, categorized by type.

4.2 Experiment 2: Performance on HTC

Table 4 presents the results of fine-tuning on the taxonomies generated by TAXMORPH. Across all evaluated benchmarks, we observe consistent improvements in classification performance when using LLM-refined taxonomies. The highest overall performance is achieved by TAXMORPH using Sonnet-3.5 in the Single Node setting, attaining an average F1 score of 0.625, compared to 0.608 for the best-performing human-curated baseline. The largest improvement occurs on the Books dataset, where the F1 score increases from 0.583 to 0.612, corresponding to a gain of +2.9 percentage points. On the Amazon dataset, the Haiku-based taxonomy achieves the best performance with an F1 score of 0.475, outperforming the baseline score of 0.457. In the WOS dataset, the gains are smaller but still notable, with Sonnet-3.5 improving the F1 score from 0.785 to 0.802.

In addition to the prompt-based approaches that leverage semantic label descriptions, we also evaluate two control conditions that exclude semantic information. The Linear Layer baseline performs worst overall, with an average F1 score of 0.403, highlighting the difficulty of learning the label space without access to textual descriptions. Using structured identifiers as label representations improves performance to 0.542, but remains substantially below any semantic variant. This result underscores the importance of label semantics for effective learning. Between the two prompt-based settings, the Single Node representation consistently outperforms the Full Path representation (e.g., 0.612 vs. 0.607 for Sonnet-3.5), suggesting that precise, well-aligned leaf-level labels are more informative than full hierarchical paths in the con-

text of TAXMORPH. This observation indicates that deeper hierarchical context may introduce redundancy or noise, which can interfere with the model's ability to learn fine-grained distinctions.

4.3 Experiment 3: Few-Shot Experiments

We evaluate the impact of TAXMORPH in few-shot settings by training models with only $k \in \{2, 4, 8, 16\}$ examples per leaf node. As previous experiment shows improvements in full fine-tuning settings, we now investigate if improved taxonomies can also help in data-scarce scenarios. All other hyperparameters and training procedures follow the setup in Section 3.1, with the exception that training is limited to 2,000 steps to reduce overfitting under low-resource conditions.

The results, shown in Figure 3, demonstrate that taxonomies refined with TAXMORPH consistently outperform human-curated taxonomies across all datasets as the number of examples increases. On the Books dataset, the F1 score improves from 0.462 (human-curated) to 0.484 with TAXMORPH (Sonnet-3.5) at $k = 16$, representing the largest observed gain in this setting. In the WOS dataset, the best-performing configuration at $k = 16$ is again TAXMORPH (Sonnet-3.5), achieving 0.691 compared to 0.667 for the human-curated hierarchy. The Amazon dataset shows smaller improvements overall, though TAXMORPH (Haiku-3.5) outperforms all other configurations at $k = 8$ and $k = 16$, reaching 0.322 and 0.323, respectively.

At $k = 2$, differences between methods are less pronounced. In Books and Amazon, the human-curated taxonomy performs comparably or slightly better than some LLM-refined versions. However, in WOS, even under this low-resource condition, TAXMORPH (Sonnet-3.5) achieves a substantial

Taxonomy	Label Representation	Amazon	Books	WOS	Avg.
Human-Curated	Linear Layer	0.142	0.348	0.720	0.403
	Structured Identifier	0.351	0.508	0.767	0.542
	Single Node	0.457	0.583	0.785	0.608
	Full Path	0.432	0.582	0.783	0.599
TAXMORPH (Haiku)	Single Node	**0.475**	0.610	0.786	0.624
	Full Path	0.457	0.603	0.790	0.617
TAXMORPH (Sonnet-3)	Single Node	0.462	0.609	0.791	0.621
	Full Path	0.439	0.602	0.789	0.610
TAXMORPH (Sonnet-3.5)	Single Node	0.460	**0.612**	**0.802**	**0.625**
	Full Path	0.447	0.607	0.796	0.617

Table 4: Results when using TAXMORPH with different LLMs. We report macro-averaged F1 scores. The final column reports the average across the three datasets.
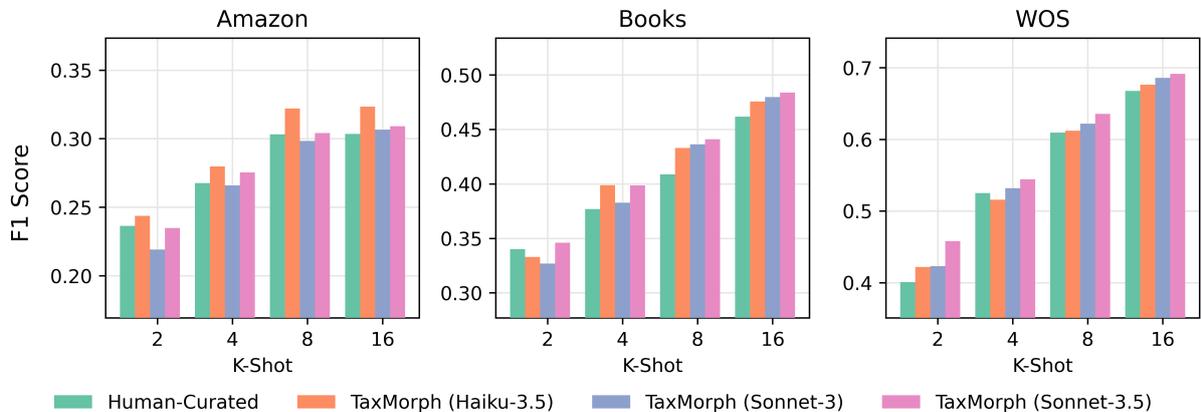

Figure 3: Few-shot performance on hierarchical text classification with human-curated versus LLM-refined taxonomies using TAXMORPH. We report macro-averaged F1 scores across four datasets using DistilBERT and the Single Node setting, varying the number of training examples per leaf node ($k \in \{2, 4, 8, 16\}$). LLM-refined taxonomies consistently outperform human-curated ones, especially as more training data becomes available.

improvement, reaching 0.458 versus 0.401 for the human-curated baseline. These results indicate that while the benefit of refinement increases with data availability, LLM-refined taxonomies can already provide gains in the low-shot regime, particularly in settings with well-structured hierarchies.

4.4 Experiment 4: Embedding Space Analysis

One central assumption is that a well-structured hierarchy should support the model in assigning input texts to their correct positions within the taxonomy. When this assignment fails, the cause may be either in the intrinsic difficulty of the input or in the taxonomy itself due to vague labels, overlapping categories, or inconsistent structure. In this ablation experiment, we investigate to what extent a language model learns a true hierarchical layout in

the embedding space and whether LLM-generated taxonomies are easier learn to learn.

Paletto et al. (2024) propose local embedding-based metrics to assess taxonomy coherence, such as measuring the similarity between sibling nodes (to which we refer as child-to-child similarity (CS) subsequently) or averaging the similarity between parents and their children (to which we refer as parent-to-child similarity (PS) subsequently). While these metrics capture surface-level structure, they do not support a deeper understanding whether a language model truely learns a hierarchical structure. To address this, we introduce the Taxonomy Probing Metric (TPM), a probing-based evaluation that directly tests whether the structure of a taxonomy is unambiguously differentiable in embedding space. To do so, TPM evaluates (1) whether
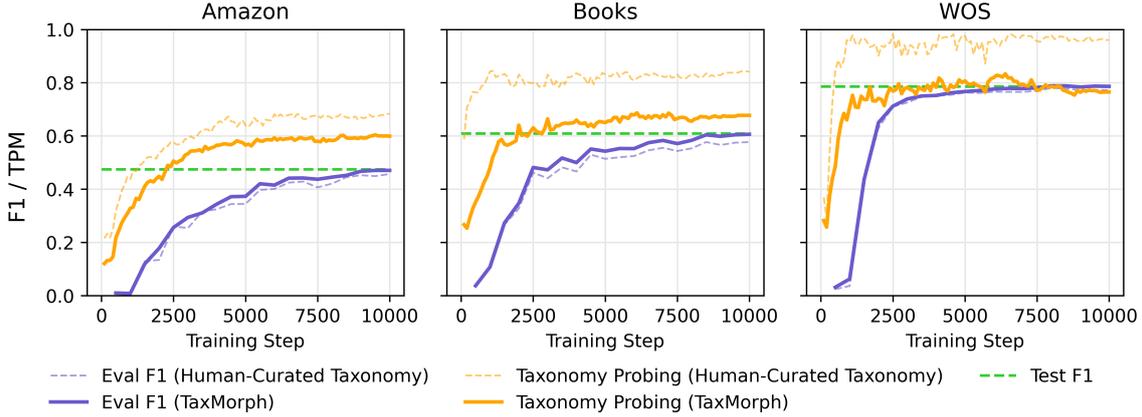
Figure 4: Comparison of F1 and our Taxonomy Probing Metric (TPM) between human-curated taxonomy and LLM-refined ones. We report the metrics for setting Prompt (Single Node) and using Haiku for refining the taxonomy.
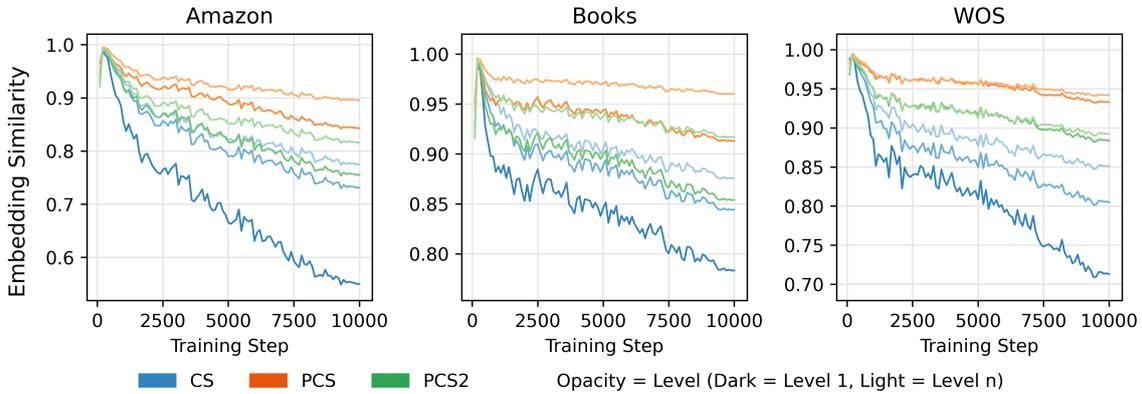

Figure 5: Semantic similarity trends of embedding-based taxonomy alignment metrics during fine-tuning. We plot Children Similarity (CS), Parent-Child Similarity (PCS), and Parent-Child Centrality (PCS2) across training steps using TAXMORPH with Haiku-3.5 and the Prompt (Single Node) setting. The metrics quantify local coherence in embedding space but do not correlate with downstream classification performance.

each child node is most similar to its correct parent among all candidate parents, and (2) whether each parent is most similar to its correct set of children among all possible child groupings. This formulation mirrors the inference process in classification: the model must choose the correct node among many. Since our setting uses taxonomy-aware prompting, TPM is designed to test whether the hierarchical structure induces meaningful and recoverable clusters in embedding space. Crucially, it provides a way to connect structural alignment to predictive behavior by asking whether the taxonomy actively supports the model in making correct decisions. We provide details on the the metric in Appendices A and B.

In Figure 4, we plot (1) the F1 score on the evaluation set during fine-tuning, (2) our proposed TPM metric, both for (3) models fine-tuned on

LLM-refined taxonomies generated by TaxMorph (using Haiku-3.5), and (4) models fine-tuned on the original human-curated hierarchy. In Figure 5, we show the alignment metrics introduced by Paletto et al. (2024) (Children Similarity (CS), Parent-Child Similarity (PCS), and Parent-Child Centrality (PCS2)) along with our additional metric PS2, which captures the cosine similarity between a parent and the average embedding of its children.

When comparing TPM to CS, PCS, and PCS2, we observe that TPM correlates much more closely with downstream F1 performance. While TPM generally increases as the model learns, the embedding-based metrics from prior work show a consistent decline during training. This trend suggests that the model pushes node representations apart in embedding space, making absolute similarity values less meaningful for evaluating structural alignment. In

contrast, TPM directly captures whether the model can differentiate parents and children among competing alternatives, aligning more with the model's decision-making objective during inference.

We further observe that TPM scores are consistently lower for LLM-refined taxonomies than for their human-curated counterparts. This suggests that while LLM-generated hierarchies lead to improved classification performance, they introduce structural complexity that makes them harder to interpret in embedding space. Qualitative analysis reveals that LLMs tend to enrich taxonomies by inserting intermediate nodes and refining label semantics, resulting in more expressive structures. These refinements improve task performance but do not necessarily yield tighter or more regular geometric organization in the embedding space. This highlights an important distinction: taxonomies that support better predictions may not be those that are easiest to embed or align spatially.

5 Related Work

Hierarchical Text Classification. Existing research leverages the general language understanding capabilities of large LMs (LLMs) for hierarchical text classification (HTC) (Zhou et al., 2020; Chen et al., 2021b; Bhambhoria et al., 2023; Wang et al., 2023). More recent works have advanced these approaches by incorporating contrastive learning techniques (Wang et al., 2022; Zhu et al., 2024) to better differentiate between similar classes, or by integrating external knowledge from knowledge graphs (Liu et al., 2023; Jain et al., 2024) to enrich the classification process. These methods typically assume that the underlying taxonomy is fixed and well-formed.

Our approach diverges from this assumption by shifting focus from classification architectures to the structure of the taxonomy itself. We treat the taxonomy not as a static input but as an object that can be refined and optimized using LLMs. This allows us to improve compatibility between the label space and the model's inductive biases, yielding gains in downstream performance even without changing the classifier. Moreover, our evaluation goes beyond task accuracy to assess structural alignment, providing insights into how changes in the hierarchy shape the learning dynamics of HTC.

Taxonomy Understanding. Beyond HTC, taxonomies play a central role in many areas of natural language processing, including taxonomy induction from unstructured text (Bosselut et al., 2019; Chen et al., 2021a), reasoning over transitive and hierarchical relations (Aly et al., 2019a; Lin and Ng, 2022; Tran et al., 2024; He et al., 2023; ?), and understanding structured knowledge representations (Wu et al., 2023; Zhou et al., 2024; Moskvoretskii et al., 2024b). Several recent works investigate whether LMs encode relational or hierarchical structure internally, often using probing tasks or synthetic benchmarks (Richard et al., 2024). Specialized models have also been proposed that explicitly encode hierarchies, including those trained with structural inductive biases (Moskvoretskii et al., 2024a) or using hyperbolic embeddings to capture the exponential growth of tree-like structures (He et al., 2024).

While these works aim to better model taxonomic structure, their evaluations often rely on local assessments, such as pairwise relation prediction (Petroni et al., 2019), masked token likelihoods, or similarity-based matching (Gao et al., 2024; Wiland et al., 2024). Our work complements and extends this literature by proposing a probing-based metric designed to reflect global alignment between model representations and hierarchical structure. Rather than evaluating alignment indirectly through static embeddings or isolated edges, we directly measure whether a given hierarchy helps the model make correct predictions.

6 Conclusion

Our study focuses on improving hierarchical text classification by refining human-curated taxonomies using large language models. We introduce TAXMORPH, a framework that leverages LLMs to generate more expressive hierarchical structures, leading to consistent performance gains across multiple HTC benchmarks. Beyond performance, we investigate how these LLM-refined taxonomies behave in embedding space. To this end, we propose the Taxonomy Probing Metric (TPM), which directly tests whether parent-child relationships remain distinguishable after fine-tuning. Our findings show that while LLM-generated hierarchies boost classification accuracy, they often reduce structural clarity in embedding space, suggesting a trade-off between expressiveness and geometric alignment. These results underscore the need to evaluate taxonomies not only by task performance but also by how well their structure aligns with model representations.

Limitations

Our study investigates the use of LLMs to refine taxonomies for hierarchical text classification. While our TAXMORPH approach yields improved downstream performance, it has several limitations:

1. The structure and content of the refined taxonomies are not yet fully explored. While the evaluation shows improved classification performance, it does not address questions related to semantic validity, potential biases, or alignment with domain-specific expectations. Additional analyses are needed to better understand the nature of the refinements produced by LLMs.

2. Although the experiments include multiple classifiers such as DistilBERT and BERT, the results are restricted to encoder-based transformer models. Other classes of models, including generative architectures, graph-based architectures or those incorporating external knowledge, are not considered. The analysis may therefore not generalize to alternative modeling choices.

3. Additional limitations include the exclusive use of LLMs from a single family (Anthropic Claude), the reliance on fixed prompting templates, and the absence of human evaluation to assess the quality of taxonomy edits. These constraints may limit the interpretability and scope of the findings.

Future work should investigate a broader range of model architectures, refinement strategies, and evaluation protocols to assess the structural properties and practical implications of LLM-generated taxonomies.

References

Rami Aly, Shantanu Acharya, Alexander Ossa, Arne Köhn, Chris Biemann, and Alexander Panchenko. 2019a. Every child should have parents: A taxonomy refinement algorithm based on hyperbolic term embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4811–4817, Florence, Italy. Association for Computational Linguistics.

Rami Aly, Steffen Remus, and Chris Biemann. 2019b. Hierarchical multi-label classification of text with capsule networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 323–330, Florence, Italy. Association for Computational Linguistics.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Shunting Zhang, Michael Suo, Phil Tillet, Xu Zhao, Eikan Wang, Keren Zhou, Richard Zou, Xiaodong Wang, Ajit Mathews, William Wen, Gregory Chanan, Peng Wu, and Soumith Chintala. 2024. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS '24, page 929–947, New York, NY, USA. Association for Computing Machinery.

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, Baltimore, Maryland. Association for Computational Linguistics.

Rohan Bhambhoria, Lei Chen, and Xiaodan Zhu. 2023. A simple and effective framework for strict zero-shot hierarchical classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1782–1792, Toronto, Canada. Association for Computational Linguistics.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Catherine Chen, Kevin Lin, and Dan Klein. 2021a. Constructing taxonomies from pretrained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4687–4700, Online. Association for Computational Linguistics.

Haibin Chen, Qianli Ma, Zhenxi Lin, and Jiangyue Yan. 2021b. Hierarchy-aware label semantics matching network for hierarchical text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th*

International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4370–4379, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Inderjit S. Dhillon, Subramanyam Mallela, and Rahul Kumar. 2002. Enhanced word clustering for hierarchical text classification. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 191–200, New York, NY, USA. Association for Computing Machinery.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Yuan He, Jiaoyan Chen, Ernesto Jimenez-Ruiz, Hang Dong, and Ian Horrocks. 2023. Language model analysis for ontology subsumption inference. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3439–3453, Toronto, Canada. Association for Computational Linguistics.

Yuan He, Zhangdie Yuan, Jiaoyan Chen, and Ian Horrocks. 2024. Language models as hierarchy encoders. *Preprint*, arXiv:2401.11374.

Nicolas Heist and Heiko Paulheim. 2021. Information extraction from co-occurring similar entities. In *Proceedings of the Web Conference 2021*, WWW '21, page 3999–4009, New York, NY, USA. Association for Computing Machinery.

Vidit Jain, Mukund Rungta, Yuchen Zhuang, Yue Yu, Zeyu Wang, Mu Gao, Jeffrey Skolnick, and Chao Zhang. 2024. HiGen: Hierarchy-aware sequence generation for hierarchical text classification. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1354–1368, St. Julian's, Malta. Association for Computational Linguistics.

Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S. Gerber, and Laura E. Barnes. 2017. Hdltex: Hierarchical deep learning for text classification. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 364–371.

Zornitsa Kozareva and Eduard Hovy. 2010. A semi-supervised method to learn and construct taxonomies using the web. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1110–1118, Cambridge, MA. Association for Computational Linguistics.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, S. Auer, and Christian Bizer. 2015. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6:167–195.

Ruixi Lin and Hwee Tou Ng. 2022. Does BERT know that the IS-a relation is transitive? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 94–99, Dublin, Ireland. Association for Computational Linguistics.

Ye Liu, Kai Zhang, Zhenya Huang, Kehang Wang, Yanghai Zhang, Qi Liu, and Enhong Chen. 2023. Enhancing hierarchical text classification through knowledge graph integration. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5797–5810, Toronto, Canada. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Viktor Moskvoretskii, Ekaterina Neminova, Alina Lobanova, Alexander Panchenko, and Irina Nikishina. 2024a. TaxoLLaMA: WordNet-based model for solving multiple lexical semantic tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2331–2350, Bangkok, Thailand. Association for Computational Linguistics.

Viktor Moskvoretskii, Alexander Panchenko, and Irina Nikishina. 2024b. Are large language models good at lexical semantics? a case of taxonomy learning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1498–1510, Torino, Italia. ELRA and ICCL.

Lorenzo Paletto, Valerio Basile, and Roberto Esposito. 2024. Label augmentation for zero-shot hierarchical text classification. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7697–7706, Bangkok, Thailand. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

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, Hong Kong, China. Association for Computational Linguistics.

Ange Richard, Laura Cristina Alonzo Canul, and François Portet. 2024. FRACAS: a FRench annotated corpus of attribution relations in newS. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7417–7428, Torino, Italia. ELRA and ICCL.

Susanna Rücker and Alan Akbik. 2023. CleanCoNLL: A nearly noise-free named entity recognition dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8628–8645, Singapore. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Aixin Sun and Ee-Peng Lim. 2001. Hierarchical text classification and evaluation. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 521–528.

Manuel Tran, Yashin Dicente Cid, Amal Lahiani, Fabian Theis, Tingying Peng, and Eldad Klaiman. 2024. Training transitive and commutative multimodal transformers with loretta. *Advances in Neural Information Processing Systems*, 36.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledge base. *Communications of the ACM*, 57:78–85.

Yue Wang, Dan Qiao, Juntao Li, Jinxiong Chang, Qishen Zhang, Zhongyi Liu, Guannan Zhang, and Min Zhang. 2023. Towards better hierarchical text classification with data generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7722–7739, Toronto, Canada. Association for Computational Linguistics.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. CrossWeigh: Training named entity tagger from imperfect annotations. 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 5154–5163, Hong Kong, China. Association for Computational Linguistics.

Zihan Wang, Peiyi Wang, Lianzhe Huang, Xin Sun, and Houfeng Wang. 2022. Incorporating hierarchy into text encoder: a contrastive learning approach for hierarchical text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7109–7119, Dublin, Ireland. Association for Computational Linguistics.

Jacek Wiland, Max Ploner, and Alan Akbik. 2024. BEAR: A unified framework for evaluating relational knowledge in causal and masked language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2393–2411, Mexico City, Mexico. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Weiqi Wu, Chengyue Jiang, Yong Jiang, Pengjun Xie, and Kewei Tu. 2023. Do PLMs know and understand ontological knowledge? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3080–3101, Toronto, Canada. Association for Computational Linguistics.

Li Yujian and Liu Bo. 2007. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095.

Jie Zhou, Chunping Ma, Dingkun Long, Guangwei Xu, Ning Ding, Haoyu Zhang, Pengjun Xie, and Gongshen Liu. 2020. Hierarchy-aware global model for hierarchical text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1106–1117, Online. Association for Computational Linguistics.

Li Zhou, Taelin Karidi, Nicolas Garneau, Yong Cao, Wanlong Liu, Wenyu Chen, and Daniel Hershcovich. 2024. Does mapo tofu contain coffee? probing llms for food-related cultural knowledge. *Preprint*, arXiv:2404.06833.

He Zhu, Junran Wu, Ruomei Liu, Yue Hou, Ze Yuan, Shangzhe Li, Yicheng Pan, and Ke Xu. 2024. HILL: Hierarchy-aware information lossless contrastive learning for hierarchical text classification. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4731–4745, Mexico City, Mexico. Association for Computational Linguistics.

A   Measuring Taxonomy Alignment

To assess the quality of refined taxonomies beyond downstream task performance, we evaluate how well language models capture the underlying hierarchical structure. We adopt embedding-based metrics proposed by Paletto et al. (2024) that quantify local taxonomic coherence in the model's representation space. Let θ denote a sentence encoder (e.g., a sentence transformer (Reimers and Gurevych, 2019)) and let $\varphi(\cdot, \cdot)$ denote cosine similarity.

Given a parent node v_i and its set of direct children, we define $V_i = \{v_i\} \cup \text{children}(v_i)$. The **Children Similarity (CS)** metric computes the average pairwise cosine similarity among all children of a given parent:

$$CS(V_i) = \frac{\sum_{v_j, v_k \in \text{children}(v_i)} \varphi(\theta(v_j), \theta(v_k))}{\binom{|\text{children}(v_i)|}{2}}.$$

The **Parent-Child Similarity (PCS)** metric measures how close a parent is to each of its children in the embedding space:

$$PCS(V_i) = \frac{\sum_{v_j \in \text{children}(v_i)} \varphi(\theta(v_i), \theta(v_j))}{|\text{children}(v_i)|}$$

We further consider the alternative **Parent-Child Centrality (PCS2)** metric as the similarity between a parent node and the average embedding of its children:

$$PCS2(V_i) = \varphi\left(\theta(v_i), \overline{\theta}(\text{children}(v_i))\right)$$

where

$$\overline{\theta}(\text{children}(v_i)) = \frac{\sum_{v_j \in \text{children}(v_i)} \theta(v_j)}{|\text{children}(v_i)|}.$$

These metrics quantify how locally consistent a taxonomy is in the embedding space. However, they are limited to pairwise or cluster-level relationships and may not fully capture structural confusion in the global hierarchy. In particular, dense clusters or weakly separated branches can result in misleadingly high scores despite poor alignment with the intended taxonomy structure.

B   Taxonomy Alignment Metric

To evaluate a model's understanding of the full hierarchical structure, we introduce a probing-based approach that measures alignment between nodes in the taxonomy and their embeddings. This method captures global structural coherence rather than local similarity alone. We use the terminology

- Let V^l be the set of nodes at level l (the children).

- Let V^{l-1} be the set of nodes at level $l-1$ (the parents).

- Let $\text{children}(v)$ be the set of children of node v.

- Let $\theta(v)$ be the embedding of a node v.

- Let $\varphi(\cdot, \cdot)$ be the cosine similarity function.

Top-down matching. This probing method evaluates whether the model can correctly associate each parent node with its corresponding set of children. The core assumption is that the average embedding of a set of child nodes should reflect the embedding of their common parent.

Let V^l denote the set of nodes at level l, and let $\text{children}(v_k^l)$ denote the set of children for a given parent node v_k^l at level l. For each parent node $v_i^l \in V^l$, we compute cosine similarity between its embedding and the average embedding of all child sets at level $l+1$. The model is then tasked with matching each parent to the most similar child set:

$$M = \{(v_i^l, \text{children}(v_{k*}^l)) \mid$$
$$k^* = \arg\max_k \varphi(\theta(v_i^l), \overline{\theta}(\text{children}(v_k^l)))\}$$

where M is the resulting set of predicted parent–child set associations. A prediction is considered correct if the assigned child set exactly matches the true children of the parent. We report top-down accuracy (TD) as the proportion of correctly matched pairs in M across all levels of the hierarchy.

Bottom-up matching. This probing method complements the previous one by verifying whether each child is assigned to its correct parent. Let V^{l-1} denote the set of parent nodes at level $l-1$, and V^l the children at level l. For each parent v_i^{l-1}, we compute a decision boundary $D(v_i^{l-1})$, defined as the minimum similarity between the parent and any of its children:

$$D(v_i^{l-1}) = \min_{v_j \in \text{children}(v_i^{l-1})} \varphi(\theta(v_i^{l-1}), \theta(v_j)).$$

Given a child $v_j^l \in V^l$, we predict its parent as the node v_k^{l-1} with highest cosine similarity:

$$k = \arg\max_{v_i^{l-1} \in V^{l-1}} \varphi(\theta(v_j^l), \theta(v_i^{l-1})).$$

Taxonomy	Label Representation	Amazon	Books	WOS	Avg.
Human-Curated	Linear Layer	0.009	0.247	0.093	0.116
	Structured Identifier	0.322	0.518	0.762	0.534
	Single Node	0.455	0.582	0.783	0.607
	Full Path	0.457	0.594	0.778	0.610
TAXMORPH (Haiku)	Single Node	**0.477**	0.591	0.793	0.620
	Full Path	0.457	0.610	0.789	0.619
TAXMORPH (Sonnet-3)	Single Node	0.466	0.611	**0.794**	0.624
	Full Path	0.464	0.618	0.786	0.623
TAXMORPH (Sonnet-3.5)	Single Node	0.471	**0.616**	0.798	**0.628**
	Full Path	0.467	0.618	0.793	0.626

Table 5: Main results when using BERT trained on LLM-refined hierarchies with TAXMORPH. We report macro-averaged F1 scores. The final column shows the average across Amazon, Books, and WOS.

The prediction is counted as correct only if: (1) v_k^{l-1} is the true parent, (2) the similarity exceeds the decision boundary $D(v_k^{l-1})$, and (3) the similarity does not exceed the boundary of any other parent. We denote the resulting accuracy as BU.

Finally, we define the Taxonomy Probing Metric as the harmonic mean of top-down and bottom-up matching scores:

$$\text{TPM}(\mathcal{G}) = \frac{2 \cdot TD \cdot BU}{TD + BU}.$$

This metric captures both local consistency and global structural understanding of the taxonomy. It rewards models that both preserve the intended hierarchy and avoid ambiguous assignments in the embedding space.

C   Results Using Different Transformers

In this section, we present the main results obtained using BERT (Devlin et al., 2019). The experimental setup is identical to the one described in the main section.

The results are consistent with those in the main section, suggesting that taxonomies refined with TAXMORPH are also effective for other student models, including those that are more powerful. However, we encountered some training instabilities with our hyperparameter configuration in the Linear Layers and Simple Enumerations settings, as the results for certain datasets were significantly lower compared to those obtained with DistilBERT.

D   Pseudocode for Counting Transformations

In Algorithm 1, we present the pseudocode used to count and classify the transformations applied by TAXMORPH. Each original path is compared to its refined counterpart to determine whether it was renamed, rearranged, merged, or extended with new nodes. This classification allows us to analyze how TAXMORPH modifies the taxonomy.

Algorithm 1 Taxonomy Transformation Classification

Require: Mapping $f : \mathcal{P} \rightarrow \mathcal{P}'$ from original to generated paths
1: **for all** $p \in \mathcal{P}$ **do**
2:    **if** $p \notin \text{dom}(f)$ **then**
3:       invalid
4:    **else**
5:       Let $p = (c_1, \ldots, c_n)$, $f(p) = (c'_1, \ldots, c'_m)$
6:       **if** $(c_1, \ldots, c_n) = (c'_1, \ldots, c'_m)$ **then**
7:          unchanged
8:       **else if** $n = m$ **then**
9:          renamed (count each $c_i \neq c'_i$)
10:      **else if** $n < m$ **then**
11:         generated (+ renamed/rearranged if endpoints differ)
12:      **else if** $n > m$ or f not injective **then**
13:         merged
14:      **end if**
15:    **end if**
16: **end for**

E Full Prompt for TAXMORPH

You are an expert taxonomist. Your task is to refine a given taxonomy represented as a list of slash-separated hierarchical paths. Each path is a string encoding a sequence of categories from root to leaf, such as `"electronics/computers/laptops"`.

Your goal is to produce a refined taxonomy by transforming these paths. Valid transformation types include:

- **Renaming** categories (e.g., `"pet supplies"` → `"pets"`)

- **Rearranging** levels (e.g., moving a category under a more appropriate parent)

- **Generating** intermediate categories to increase granularity

- **Merging** semantically redundant categories

You are not allowed to generate new, additional leaf nodes. Only modify a path if there is a clear reason. Output must be in **strict JSON** format: a dictionary mapping each original path to its refined version. Both keys and values must be strings using the same slash-separated format.

Input taxonomy (example):

```
[
  "electronics/computers/laptops",
  "grocery/fruit/citrus/orange",
  "grocery/beverages/soft drinks",
  ...
]
```

Expected output format:

```
{
  "electronics/computers/laptops":
  "electronics/computing/laptops",
  ...
}
```

F Expressiveness vs. Alignment

In Figure 6, we present the same training dynamics as in Figure 4, but using more expressive label semantics by providing the full path during fine-tuning. Surprisingly, performance decreases despite the increased expressiveness. Although the model can, in principle, learn parent-child relationships in this setting, it becomes more easily confused. This suggests that taxonomies are best understood by language models when labels at each level are clear and discriminative.
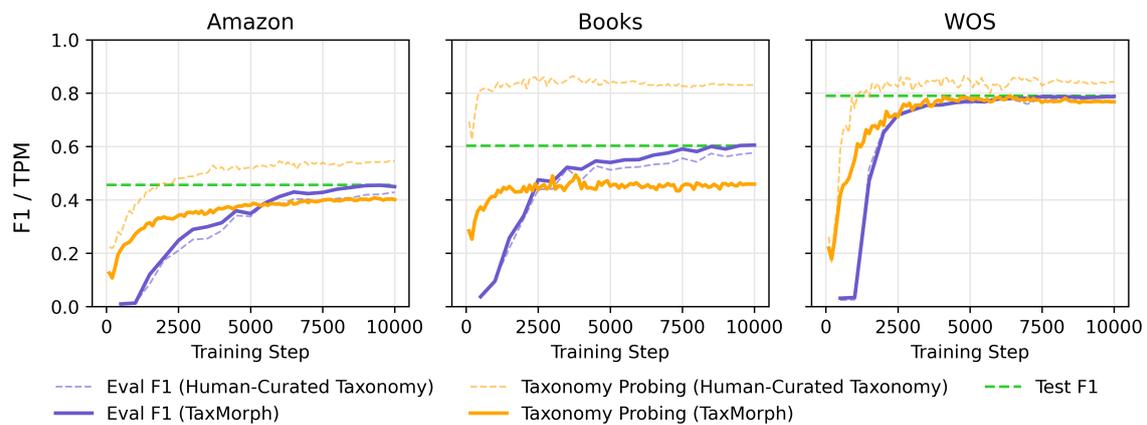
Figure 6: Comparison of F1 and our Taxonomy Probing Metric (TPM) between human-curated taxonomy and LLM-refined ones. We report the metrics using the entire hierarchy path as label semantics. We use Haiku-3.5 for refining the taxonomy.