PaCoClass: Leveraging Path Selection and Conditional Classifier for Hierarchical Text Classification with Minimal Supervision

Anonymous ACL submission

Abstract

Hierarchical Text Classification (HTC) aims to categorize documents into specific paths within a label taxonomy. Most existing works for this task are fully supervised methods, which demand substantial time and effort from domain experts for data annotation. On the other hand, only a handful of studies have focused on HTC with severely constrained supervision signals. They usually adopt a top-down strategy to identify document-relevant classes, yet risk missing some truly essential candidates. And they often adopt a flat classifier architectures, which probably produce predictions violating hierarchical constraint. To address these limitations, we propose a novel weakly-supervised HTC framework called PaCoClass. Specifically, PaCoClass first introduces a Bidirectional Path Consistency Scoring mechanism to quantify document-label path semantic alignment by combining bottom-up candidates retrieval with top-down consistency constraints. Subsequently, PaCoClass designs an LLM-Enhanced Path Refinement strategy, which introduces Large Language Models (LLMs) to further refine high scoring paths. Thirdly, a conditional classifier architecture is introduced instead of flat classifiers, which inherently enforces hierarchical constraints and captures intrinsic label dependencies. Finally, experiments demonstrate that our framework consistently outperforms several well-known competing HTC methods at all stages.

1 Introduction

Hierarchical Text Classification (HTC) (Sun and Lim, 2001) aims to assign a set of classes from a label taxonomy to documents. These classes cover concepts ranging from broad to specific and are linked by parent-child relationships, forming paths that lead from a root node to specific class. For instance, in Amazon product review classification, a document might be assigned to the path "Beauty → Hair Care → Shampoo", where each level provides progressively finer-grained semantic distinctions. Compared to flat text classification, which treats all categories as independent labels, HTC explicitly models the hierarchical relationships among classes, enabling more precise categorization and more effective organization of large-scale document collections (Zangari et al., 2024). This capability makes HTC indispensable in numerous real-world applications, including scientific literature indexing (Zhang et al., 2023) and e-commerce product categorization. Meanwhile, hierarchically structured text (Huang et al., 2025; Jiao et al., 2025) also plays an important role in retrieval-augmented generation for Large Language Models (LLMs).

Most existing HTC methods adopt fully supervised learning paradigms (Im et al., 2023; Kim et al., 2024), which depend on large-scale human-annotated data for model training. Despite their success, applying these methods to real-world scenarios remains challenging (Wang et al., 2023) due to costly, time-consuming manual annotation—an issue especially critical for HTC, as its hierarchical taxonomy demands far greater annotation efforts from domain experts.

To alleviate the annotation burden, weakly-supervised hierarchical text classification (WHTC) methods have been explored. For instance, Meng et al. (Meng et al., 2019) utilize a small set of keywords or labeled documents for each class to train a text classifier. To further minimize supervision, TaxoClass (Shen et al., 2021) pioneers the minimal supervision paradigm by relying solely on class names to identify "core classes" (i.e., fine-grained classes that most accurately describe the documents) via textual entailment model. Building upon this, TELEClass (Zhang et al., 2025) further enhances the supervision signal by enriching the taxonomy with class-indicative terms derived from both LLMs and the corpus.

In addition to these WHTC methods, some research explores the strict zero-shot setting. For

instance, Bongiovanni et al. (2023) and Paletto et al. (2024) compute the similarity between the document and all classes. They utilize the label taxonomy through Upward Score Propagation (UP), which bottom-up propagates scores from child nodes to their parents to improve the classification performance for upper-level categories. However, such bottom-up strategy risks validating parent nodes solely based on high-scoring children, ignoring the potential semantic mismatch between the parent nodes and the document, as illustrated in Figure 1-(b).

Although the aforementioned methods have successfully reduced the dependency on supervision signals, they still exhibit the following limitations: (1) Core class selection dilemma: Most WHTC methods employ a top-down strategy that traverses from coarse to fine-grained levels to select candidate core class, risking premature pruning where correct fine-grained classes are excluded if their ancestors fail to attain high scores (as illustrated in Figure 1-(a)). (2) Underutilization of LLMs' classification capability: Existing methods primarily utilize LLMs for data augmentation or taxonomy enrichment, overlooking their potential to refine predictions. (3) Classifier architectural mismatch: Most WHTC methods adopt flat, hierarchy-agnostic architectures that fail to enforce structural constraints. This often results in logically inconsistent predictions, such as a child node having a higher probability than its parent.

To address these limitations, we propose **PaCo-Class** (**Pa**th Selection and **Co**nditional **Class**ifiers), a progressive framework that integrates path-based pseudo-labeling with parent-conditioned classifier training for HTC with minimal supervision, relying solely on class names. Our main contributions are as follows.

- Contrary to existing WHTC methods with top-down strategy, we propose a **Bidirectional Path Consistency Scoring (BPCS)** mechanism in HTC task. It first identifies highly relevant fine-grained classes via a bottom-up strategy, then utilizes top-down hierarchical consistency constraints to filter out paths lacking ancestral semantic support, effectively resolving the pruning errors common in top-down strategy.

- We design an **LLM-Enhanced Path Refinement** strategy that constructs a high-quality candidate set via BPCS-generated paths and lexical matches for LLM reasoning, thereby enabling LLMs to effectively refine paths.

- Contrary to flat classifier used in HTC task, we design the **Hierarchy-Aware Conditional Classifier (HACC)** that decomposes the global classification problem into recursive local predictions conditioned on parent nodes. Through dynamic parent-guided attention and local bilinear matching, HACC focuses computational resources on distinguishing hard negatives (i.e., semantically similar siblings Chen et al. (2024)) while inherently enforcing structural consistency across the taxonomy.

- In terms of the above three technical components, a progressive WHTC framework is proposed. That is, it can employ BPCS alone for zero-shot scenarios, incorporate LLM refinement for enhanced accuracy, or train the HACC when unlabeled corpora are available. Finally, experiments conducted on two benchmark HTC datasets demonstrate that PaCo-Class achieves state-of-the-art performance at each stage compared to well-known HTC methods under the same supervision setting.

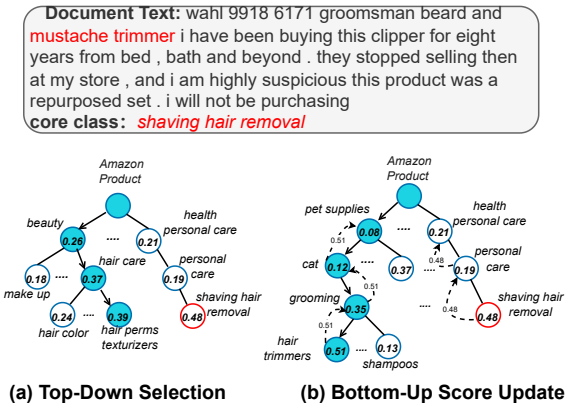

Figure 1: Two strategies for a sample document: (a) Top-Down Selection; (b) Bottom-Up Score Update. Arrows indicate selection order. The dashed line represents the direction of score updates.

2 Related Work

To alleviate the annotation burden in HTC task, some WHTC methods have been proposed. For instance, Meng et al. (Meng et al., 2019) propose WeSHClass, which utilizes a small set of keywords or labeled documents for each class to train a hierarchical neural text classifier. By generating pseudo

documents from weak supervision signals and employing self-training, WeSHClass reduces the dependency on large-scale annotations. However, compiling keyword lists for hundreds of classes and securing representative documents for niche subcategories still demand significant human efforts.

To further minimize supervision, TaxoClass (Shen et al., 2021) and TELEClass (Zhang et al., 2025) tackle HTC using only class names. Both methods centered around the concept of core classes based on the observation of human cognitive processes: experts usually first identify the most essential classes relevant to a document as core classes, and then incorporate the ancestor classes of these core classes. However, despite this motivation, TaxoClass employs a top-down strategy to reduce the search space of core classes, which may inevitably filter out correct labels due to the semantic gap between coarse-grained parent classes and specific document content.

Regarding classifier architectures, WeSHClass (Meng et al., 2019) constructs multiple local classifiers at each hierarchical node. However, this approach requires training a large number of independent classifiers. In contrast, TaxoClass and TELEClass adopt fundamentally flat, hierarchy-agnostic classifier architectures. Using a standard log-bilinear matching network optimized with Binary Cross-Entropy (BCE) loss, they treat each taxonomy node as an independent binary classification task. While this flat design simplifies training, it cannot ensure that a child class's probability does not exceed its parent's, thus placing the entire burden of learning structural dependencies on pseudo-labeled data distributions instead of embedding inductive biases into the model.

Beyond the aforementioned training-based weakly supervised methods, Zero-Shot Classification (ZSC) is particularly valuable when unlabeled corpora are unavailable or training resources are limited. ZSC methods utilize only pre-trained language models and the label taxonomy for classification. One primary approach frames the ZSC problem as a 1-Nearest Neighbor classification task based on label embeddings serving as prototypes or centroids. In this vein, Bongiovanni et al. (2023) introduced an algorithm termed Upward Score Propagation (UP), which first computes prior relevance scores between documents and labels using pre-trained language models and then re-calibrates parental scores by leveraging strong relevance signals from their corresponding children nodes. Building upon this, Paletto et al. (2024) proposed Hierarchical Label Augmentation (HiLA) to mitigate the issue of low semantic specificity at the leaf level. By leveraging LLMs to generate fine-grained child nodes for existing leaves, HiLA successfully extends the UP mechanism to the deepest level of the original taxonomy, achieving state-of-the-art performance in zero-shot HTC.

However, we identify fundamental limitations inherent to the UP algorithm mentioned above. Specifically, its update rule operates in a unidirectional reinforcement manner, that is, parent node scores are only enhanced or replaced by stronger signals from child nodes. As a result, if a major high-scoring error appears at lower levels, it will spread upward without any restriction. On the contrary, our method to be proposed in this paper will fix the above issues.

3 Method

3.1 Problem Formulation

Given a corpus $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$ with N documents and a label taxonomy $\mathcal{T}$, the taxonomy is a tree-structured hierarchy with maximum depth L and label set $\mathcal{C}$. Each label $c_j^l \in \mathcal{C}$ is indexed by its level $l \in \{0, \ldots, L\}$ and position j within that level, we use c^l or c to denote a label at level l or a generic class, where c^0 denoting the root node. Following the notations in HiLA (Paletto et al., 2024), we use $\uparrow c_j^l$ and $\Uparrow c_j^l$ to denote the parent and ancestor set of c_j^l, and $\downarrow c_j^l$ and $\Downarrow c_j^l$ for its children and descendants.

The HTC task aims to assign a subset of relevant labels $Y \subset \mathcal{C}$ to each document $d \in \mathcal{D}$, subject to the hierarchy constraint: if a label is assigned, all its ancestors must also be assigned. The output can thus be represented as label path $y = (y_1, y_2, \ldots, y_L)$, where each y_l belongs to level l of the taxonomy.

As shown in Figure 2, we propose a novel WHTC framework composed of three technical components, including Bidirectional Path Consistency Scoring (BPCS) in Section 3.2, LLM-Enhanced Path Refinement in Section 3.3, and Hierarchy-Aware Conditional Classifier (HACC) in Section 3.4. The first two components are implemented using only the label taxonomy $\mathcal{T}$ and individual documents d, producing pseudo-labels for an unlabeled corpus $\mathcal{D}$. The last component

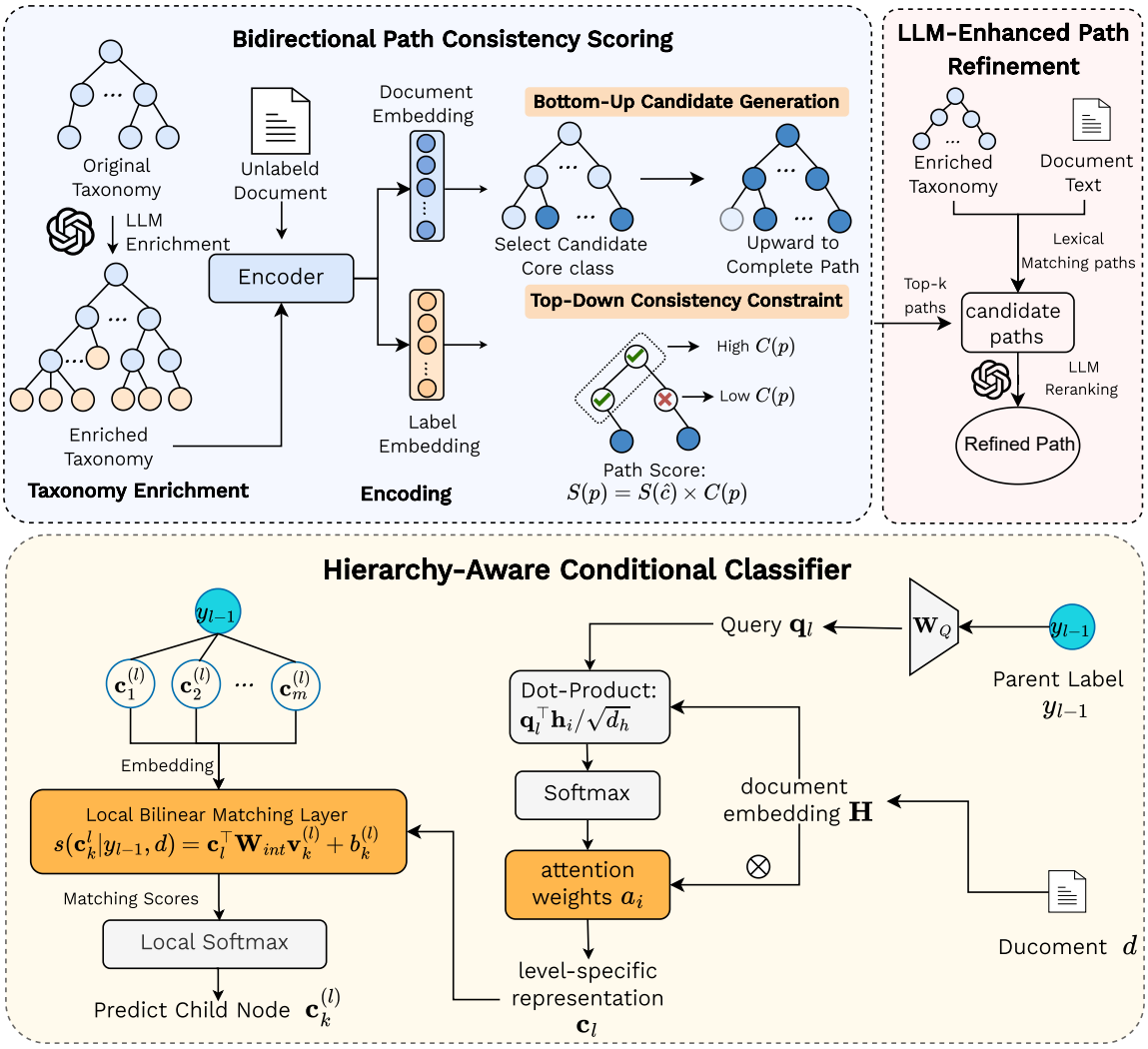

Figure 2: The overall framework of PaCoClass. The upper part illustrates the pseudo-label generation process via Bidirectional Path Consistency Scoring (BPCS) and LLM-Enhanced Path Refinement. The lower part shows the prediction process of the Hierarchy-Aware Conditional Classifier (HACC) at a specific layer.

then leverages these pseudo-labeled documents to train the conditional classifier. Technical details are as follows.

3.2 Bidirectional Path Consistency Scoring

To effectively integrate fine-grained semantic matching with global taxonomy constraints, we propose the BPCS mechanism. Unlike traditional methods that rely on unidirectional traversal, BPCS treats the document-label alignment as a bidirectional verification process. Specifically, it first employs a Bottom-Up strategy to identify all candidate core classes, ensuring that the true core classes are not prematurely pruned. Subsequently, it examines the semantic similarity between ancestor nodes and the document in a top-down manner, effectively filtering out spurious candidates that lack ancestral semantic support. Before detailing the scoring mechanism, we first introduce the necessary preparations, including taxonomy enrichment and encoding.

Taxonomy Enrichment To bridge the semantic gap between abstract class names and specific document content, we first leverage LLMs to enrich the taxonomy by generating finer-grained child nodes $\downarrow c_j^L$ for each leaf node c_j^L. The specific prompt templates are provided in Appendix B.1.

Encoding We employ a pre-trained language model to map the input document d and each label c_j^l into the same semantic vector space. For the document, we obtain the sequence of token embeddings $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_n] \in \mathbb{R}^{n \times d_h}$ and the global representation $\mathbf{h}_d \in \mathbb{R}^{d_h}$. For labels, we obtain their representations $\mathbf{v}_j^{(l)} \in \mathbb{R}^{d_h}$. The prior relevance score is computed via cosine similarity as

$S(c_j^l) = \cos(\mathbf{h}_d, \mathbf{v}_j^{(l)})$.

Building upon these preparations, the BPCS mechanism is executed through two stages, as elaborated in the subsequent sections.

3.2.1 Bottom-Up Candidate Generation

For each leaf node c_j^L in the taxonomy, we first define its leaf cluster $\mathcal{C}(c_j^L) = \{c_j^L\} \cup \downarrow c_j^L$, which contains c_j^L itself and all its LLM-generated child nodes $\downarrow c_j^L$ (see Taxonomy Enrichment in Section 3.2). From the leaf cluster onwards, we identify as core classes those classes whose prior relevance score $S(\cdot)$ is higher than the statistical threshold α, defined as the top percentile of the score distribution fitted on a large background corpus of random irrelevant documents.

$$\begin{aligned} \mathcal{C}_{core}(c_j^L) &= \{c \in \mathcal{C}(c_j^L) \mid S(c) > \alpha_c\} \\ \mathcal{A}_{core}(c_j^L) &= \{c \in \Uparrow c_j^L \mid S(c) > \alpha_c\} \end{aligned} \tag{1}$$

Based on the above sets, we determine the core classes $\hat{c}$ below.

$$\hat{c} = \begin{cases} \underset{c \in \mathcal{C}_{\text{core}}(c_j^L)}{\arg\max}\ S(c) & \text{if } \mathcal{C}_{\text{core}}(c_j^L) \neq \emptyset \\ \underset{c \in \mathcal{A}_{\text{core}}(c_j^L)}{\arg\max}\ \text{level}(c) & \text{else if } \mathcal{A}_{\text{core}}(c_j^L) \neq \emptyset \\ \text{None} & \text{otherwise} \end{cases} \tag{2}$$

We prioritize core classes in the leaf cluster $\mathcal{C}(c_j^L)$ for their specific semantic distinctions; failing that, we select the deepest qualifying ancestor from $\mathcal{A}_{core}(c_j^L)$. This bottom-up strategy avoids the top-down filtering pitfalls where semantically fine-grained core classes may be prematurely excluded due to their coarse-grained parents failing to meet relevance thresholds. Furthermore, this approach better aligns with the fundamental principle of core class identification, namely selecting the fine-grained classes that most accurately describe the documents.

3.2.2 Top-Down Consistency Constraint

To filter out spurious candidates lacking ancestral support, we introduce a path scoring mechanism that incorporates global structural consistency. For a candidate core class $\hat{c}$ and its path p, the final score $S(p)$ combines the bottom-up relevance $S(\hat{c})$ with a top-down hierarchy consistency factor $C(p)$:

$$S(p) = S(\hat{c}) \times C(p) \tag{3}$$

$C(p)$ aggregates the validity of all ancestor nodes $\Uparrow \hat{c}$. We define the validation weight w_{c_i} for each ancestor c_i as:

$$w_{c_i} = \sigma(k \cdot (S(c_i) - \alpha_{c_i})) \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function and k regulates sensitivity. w_{c_i} approaches 1 when the ancestor is relevant ($S(c_i) > \alpha_{c_i}$) and 0 otherwise. We compute $C(p)$ as the geometric mean of these weights:

$$C(p) = \left(\prod_{c_i \in \Uparrow \hat{c}} w_{c_i} \right)^{\frac{1}{|\Uparrow \hat{c}|}} \tag{5}$$

This geometric mean ensures that if any ancestor is irrelevant ($w_{c_i} \approx 0$), the entire path score is suppressed, effectively enforcing hierarchical consistency.

3.3 LLM-Enhanced Path Refinement

Although the BPCS mechanism proposed in Section 3.2 outperforms prior zero-shot methods, it essentially relies on Pre-trained Embedding Models. This reliance limits its capacity to capture complex contextual logic and distinguish subtle semantic nuances (Liu et al., 2025), which are areas where current state-of-the-art LLMs excel.

To introduce the powerful capabilities of LLMs, especially to correct potential noise in BPCS pseudo-labels, we adopt a "Recall-Rerank" two-stage architecture below. We first construct a high-quality path candidate set containing potential correct answers via two complementary strategies, and then use an LLM for screening. Specifically, we select the Top-K highest-scoring paths from BPCS as the candidate set P_{knn}, serving as a high-quality candidate pool. To complement the limited exact-matching capability of semantic retrieval, we also retrieve class names that appear in the document. Since labels have been enriched by LLMs through the Taxonomy Enrichment step, they form a keyword set containing finer-grained labels and relevant instances. All matched label paths are aggregated into P_{key}.

The above two strategies form an effective complement: **Semantic Retrieval** focuses on evaluating the overall implicit similarity of the document, fully utilizing the pseudo-labeling knowledge of BPCS. While **Lexical Matching** captures explicit lexical features in the text. We take the union of the results of both to form the final candidate path set $P_{gen} = P_{knn} \cup P_{key}$.

In the final Reranking stage, we use an LLM to select the optimal path from P_{gen}. We construct

a structured Prompt, inputting the full text of the document together with P_{gen} into the LLM. The specific prompt template is detailed in Appendix B.2. In this way, we successfully deeply integrate the reasoning capability of the LLM with the hierarchical text classification task, further improving the classification performance based on BPCS pseudo-labeling.

3.4 Hierarchy-Aware Conditional Classifier (HACC)

Most existing WHTC methods employ flat classifier architectures that predict all label scores simultaneously. However, such architectures face two major limitations. Specifically, they often ignore hierarchical dependencies, which leads to logically invalid paths where child node probabilities exceed those of their parents. Furthermore, global optimization in these models is frequently hindered by numerous easy negatives from unrelated branches, diluting the focus on hard negatives and resulting in insufficient fine-grained feature learning.

To address these limitations, we propose the Hierarchy-Aware Conditional Classifier (HACC). HACC decomposes the global classification task into a sequence of local predictions, where each step is conditioned on the identified parent node. This architecture inherently enforces hierarchical constraints and directs the model to focus on distinguishing hard negatives among sibling classes.

3.4.1 Model Architecture

Building upon the encoding defined in Section 3.2, HACC introduces two key components as follows.

Parent-Guided Attention We extract hierarchy-specific semantic features from the document. For the first level ($l = 1$), we directly use the global document representation $\mathbf{h}_d$ as the level-specific representation $\mathbf{c}_1$. For deeper levels ($l > 1$), we employ a parent-guided attention mechanism. Let y_{l-1} be the parent node. We compute a query vector $\mathbf{q}_l$ by projecting the parent embedding $\mathbf{v}_{y_{l-1}}$:

$$\mathbf{q}_l = \mathbf{W}_Q \mathbf{v}_{y_{l-1}} \tag{6}$$

This query is used to compute attention weights a_i over the document tokens $\mathbf{H}$ to obtain the level-specific representation $\mathbf{c}_l$:

$$\begin{aligned} a_i &= \frac{\exp(\mathbf{q}_l^\top \mathbf{h}_i / \sqrt{d_h})}{\sum_{k=1}^{n} \exp(\mathbf{q}_l^\top \mathbf{h}_k / \sqrt{d_h})} \\ \mathbf{c}_l &= \sum_{i=1}^{n} a_i \mathbf{h}_i \end{aligned} \tag{7}$$

By focusing on semantic segments relevant to the current parent category, this mechanism generates a level-specific representation $\mathbf{c}_l$ to facilitate more accurate local classification.

Local Bilinear Matching We strictly limit the prediction scope to the valid child node set $\downarrow y_{l-1} = \{c_1^l, c_2^l, \ldots, c_m^l\}$, where m is the number of children. We compute the matching score between $\mathbf{c}_l$ and the k-th candidate child node $c_k^l \in \downarrow y_{l-1}$ through a bilinear transformation:

$$s(c_k^l | y_{l-1}, d) = \mathbf{c}_l^\top \mathbf{W}_{int} \mathbf{v}_k^{(l)} + b_k^{(l)} \tag{8}$$

where $\mathbf{W}_{int} \in \mathbb{R}^{d_h \times d_h}$ is a learnable interaction matrix, and $b_k^{(l)}$ is a category-specific bias term. This local matching design naturally concentrates the optimization focus on semantically similar sibling nodes (hard negatives), avoiding the problem of gradient directions being dominated by a large number of irrelevant categories (easy negatives) in the early stages of training with global classifiers.

3.4.2 Model Training

We train HACC using the pseudo-label paths obtained from the LLM-Enhanced Path Refinement stage (Section 3.3) with a Teacher Forcing strategy.

By decomposing the global optimization into local sub-problems within $\downarrow y_{l-1}$, HACC eliminates easy negatives and focuses on distinguishing hard negatives (sibling nodes). This design also inherently enforces structural consistency, as child nodes are only predicted conditional on their parents.

We minimize the Local Cross-Entropy Loss:

$$\mathcal{L}_{local}(l) = -\log \frac{\exp(s(y_l | y_{l-1}, d))}{\sum_{c_k^l \in \downarrow y_{l-1}} \exp(s(c_k^l | y_{l-1}, d))} \tag{9}$$

For a pseudo-labeled path of depth k for document d, we define the document-level total loss as the sum of local losses:

$$\mathcal{L}_{total}(d) = \sum_{l=1}^{k} \mathcal{L}_{local}(l). \tag{10}$$

4 Experimental Setup

Datasets We evaluate our methods on two benchmark datasets: **Amazon-531**(McAuley and Leskovec, 2013), containing product reviews from Amazon, and **DBPedia-298**(Lehmann et al., 2015), built from Wikipedia articles. Table 4 summarizes their statistics. Following standard practice, we perform pseudo-labeling and classifier training on

Supervision	Method	Amazon-531			DBPedia-298		
		L. 1	L. 2	L. 3	L. 1	L. 2	L. 3
Zero-Shot	UP	0.7060	0.3347	0.1766	0.7846	0.6618	0.6034
	HiLA	0.7790	0.3668	0.2459	0.8064	0.6349	0.5695
	BPCS	0.8068	0.3935	0.2570	0.8490	0.7007	0.6215
	LLM+BPCS	**0.9006**	**0.5737**	**0.3755**	**0.9502**	**0.8487**	**0.8378**

Table 1: Strict Zero-shot performance comparison. Best scores in bold.

Supervision	Method	Amazon-531				DBPedia-298			
		Example-F1	P@1	P@3	MRR	Example-F1	P@1	P@3	MRR
Weakly-Supervised	WeSHClass†	0.2458	0.5773	0.2517	—	0.3047	0.5359	0.3048	—
	TaxoClass-NoST†	0.5431	0.7918	0.5414	0.5911	0.7712	0.8621	0.7712	0.8221
	TaxoClass†	0.5934	0.8120	0.5894	0.6332	0.8156	0.8942	0.8156	0.8762
	TELEClass†	0.6483	0.8505	0.6421	0.6865	0.8633	0.9351	0.8633	0.8864
	PaCoClass	**0.7115**	**0.8862**	**0.7848**	—	**0.9311**	**0.9809**	**0.9311**	—

Table 2: Weakly-supervised performance comparison. Best scores in bold. † indicates results reported in prior work. Note that WeSHClass and PaCoClass cannot generate MRR metrics because they only output predictions corresponding to the specified levels.

the training set, and evaluate model performance on the test set.

Competing Methods We compare PaCoClass with the following state-of-the-art baselines. For zero-shot settings, we include **UP** (Bongiovanni et al., 2023), which propagates relevance scores from children to parents, and **HiLA** (Paletto et al., 2024), which augments leaf nodes with LLM-generated children. For weakly-supervised settings, we compare against: **WeSHClass** (Meng et al., 2019), which generates pseudo-documents for classifier training; **TaxoClass** (Shen et al., 2021), which identifies core classes via textual entailment; and **TELEClass** (Zhang et al., 2025), which enriches the taxonomy with class-indicative terms.

Implementation Details We use Sentence Transformer (Reimers and Gurevych, 2019) `all-mpnet-base-v2` as the encoder for the similarity measure in Section 3.2 and 3.3. For the classifier training, we use `BERT-base-uncased`(Devlin et al., 2019) to ensure a fair comparison with baselines. For the LLM-Enhanced Path Refinement stage, we employ GPT-3.5-turbo as the reasoning model.

5 Experimental Results

We evaluate PaCoClass across its three progressive stages and compare against state-of-the-art baselines in both zero-shot and weakly-supervised settings. For the zero-shot setting, we report Micro-F1 scores at each hierarchy level following Bongiovanni et al. (2023). For the weakly-supervised setting, we adopt Example-F1, Precision at k (P@k) where $k \in \{1, 3\}$, and Mean Reciprocal Rank (MRR) as evaluation metrics, consistent with prior work (Shen et al., 2021).

Zero-Shot Performance Table 1 presents the zero-shot classification results. Our BPCS method outperforms the UP and HiLA baselines across all hierarchy levels on both datasets, with more substantial improvements at higher levels (L. 1, L. 2). This validates our bidirectional path consistency scoring mechanism: via the consistency factor $C(p)$, BPCS suppresses spurious high-scoring leaf nodes lacking ancestral semantic support. It effectively combines bottom-up specificity with bidirectional consistency checks.

The integration of LLMs (LLM+BPCS) yields a dramatic performance boost, significantly outpacing standalone BPCS: on DBPedia-298 alone, the gains escalate to +14.80 at L. 2 and +21.63 at L. 3. This trend demonstrates that LLMs excel at disambiguating fine-grained semantic distinctions when provided with a constrained, high-quality candidate set generated by BPCS, effectively addressing both context window limitations and attention dilution issues inherent in applying LLMs to large-scale taxonomies.

Weakly-Supervised Performance Table 2 compares PaCoClass against weakly-supervised base-

Method	Amazon-531				DBPedia-298			
	Example-F1	P@1	P@3	MRR	Example-F1	P@1	P@3	MRR
w/o LLM Enhanced	0.5997	0.8250	0.5942	—	0.8367	0.9357	0.8367	—
w/o HACC	0.6349	0.8265	0.6294	0.6661	0.8927	0.9481	0.8927	0.9090
PaCoClass	**0.7115**	**0.8862**	**0.7848**	—	**0.9311**	**0.9809**	**0.9311**	—

Table 3: Performance of PaCoClass and its ablations. The best score is boldfaced.

Statistics	Amazon-531	DBPedia-298
Hierarchy Levels	3	3
# Classes (L. 1/L. 2/L. 3)	6/64/510	9/70/219
# Training Samples	29,487	196,665
# Test Samples	19,685	49,167

Table 4: Dataset statistics.

lines. PaCoClass consistently achieves the best performance across all metrics on both datasets. We attribute this superiority to two key factors. First, our BPCS mechanism combined with LLM refinement generates higher-quality pseudo-labels than previous methods. Second, HACC, which focuses on sibling discrimination, effectively learns fine-grained distinctions and thus yields more accurate predictions.

Ablation Study Table 3 validates the contribution of each component in PaCoClass. Specifically, w/o LLM Enhanced directly utilizes the pseudo-labels obtained from Section 3.2, and w/o HACC employs the flat classifier from TaxoClass (Shen et al., 2021). Removing LLM-enhanced refinement (w/o LLM Enhanced) causes substantial drops across both datasets, where Example-F1 decreases by 11.18 points on Amazon-531 and 9.44 points on DBPedia-298, confirming that LLM verification effectively corrects embedding-based retrieval errors and filters noisy pseudo-labels. Removing the conditional classifier (w/o HACC) while retaining LLM refinement still results in 7.66-point and 3.84-point drops respectively, demonstrating that HACC's parent-conditioned local prediction provide additional gains. Notably, the relative contribution of HACC is smaller on DBPedia-298, likely because its simpler taxonomy structure reduces the complexity of hierarchical decision-making. These ablations confirm that both LLM refinement and HACC's architectural design are essential for achieving state-of-the-art performance, with their benefits amplifying as taxonomy scale and depth increase.

These results demonstrate two key strengths of our approach. First, our BPCS-based pseudo-labeling provides higher-quality training signals. Second, our HACC architecture decomposes the global classification task into parent-context-conditioned local decisions, thereby focusing optimization on distinguishing sibling nodes.

6 Conclusions

In this paper, we proposed PaCoClass, a progressive framework designed to tackle the challenges of hierarchical text classification with minimal supervision. By identifying and addressing the limitations of existing methods, namely the candidate pruning error in top-down strategy and the structural inconsistency in flat classifiers, PaCoClass introduces a robust pipeline comprising Bidirectional Path Consistency Scoring (BPCS), LLM-Enhanced Path Refinement, and a Hierarchy-Aware Conditional Classifier (HACC).

Our approach effectively combines the semantic specificity of bottom-up retrieval with the structural rigor of top-down constraints, while further leveraging the reasoning power of LLMs to refine predictions. Furthermore, the proposed conditional classifier inherently enforces hierarchical dependencies, ensuring logically valid outputs. Extensive experiments on benchmark datasets demonstrate that PaCoClass significantly outperforms state-of-the-art baselines in both zero-shot and weakly-supervised settings. The framework's modular design allows for flexible adaptation to various resource constraints, offering a practical solution to real-world HTC applications where human annotation is scarce.

7 Limitations

Despite its effectiveness, our proposed framework still has the following two limitations. First, the BPCS mechanism relies on statistical thresholds (α) derived from a general-purpose background corpus. Consequently, applying PaCoClass to

highly specialized or technical datasets may necessitate domain-specific recalibration of these thresholds to ensure optimal performance in filtering spurious paths. Second, the performance of the LLM-enhanced path refinement stage is significantly influenced by the capabilities of the underlying large language model. While we utilized `gpt-3.5-turbo` in our experiments for consistency and comparison with prior work, we believe the proposed method remains effective when integrated with various open-source and freely accessible models. Further evaluation across a broader range of models is left for future work.

References

Lorenzo Bongiovanni, Luca Bruno, Fabrizio Dominici, and Giuseppe Rizzo. 2023. Zero-shot taxonomy mapping for document classification. In *The 38th ACM/SIGAPP Symposium on Applied Computing*, pages 896–903.

Huiyao Chen, Yu Zhao, Zulong Chen, Mengjia Wang, Liangyue Li, Meishan Zhang, and Min Zhang. 2024. Retrieval-style in-context learning for few-shot hierarchical text classification. *Transactions of the Association for Computational Linguistics*, 12:1214–1231.

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, MN. Association for Computational Linguistics.

Haoyu Huang, Yongfeng Huang, Junjie Yang, Zhenyu Pan, Yongqiang Chen, Kaili Ma, Hongzhi Chen, and James Cheng. 2025. Retrieval-augmented generation with hierarchical knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6044–6060, Suzhou, China. Association for Computational Linguistics.

Sanghun Im, Gibaeg Kim, Heung-Seon Oh, Seongung Jo, and Donghwan Kim. 2023. Hierarchical text classification as sub-hierarchy sequence generation. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI'23)*, pages 12933–12941.

Yihan Jiao, Zhehao Tan, Dan Yang, Duolin Sun, Jie Feng, Yue Shen, Jian Wang, and Peng Wei. 2025. Hirag: Hierarchical-thought instruction-tuning retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5111–5130, Suzhou, China. Association for Computational Linguistics.

Gibaeg Kim, SangHun Im, and Heung-Seon Oh. 2024. Hierarchy-aware biased bound margin loss function for hierarchical text classification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7672–7682.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. Dbpedia – a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Yuxiang Liu, Tian Wang, Gourab Kundu, Tianyu Cao, Guang Cheng, Zhen Ge, Jianshu Chen, Qingjun Cui, and Trishul Chilimbi. 2025. Exploring reasoning-infused text embedding with large language models for zero-shot dense retrieval. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25)*, pages 4981–4985, New York, NY, USA. ACM.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 165–172.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2019. Weakly-supervised hierarchical text classification. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI'19)*, page 838.

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.

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.

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.

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

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.

Alessandro Zangari, Matteo Marcuzzo, Matteo Rizzo, Lorenzo Giudice, Andrea Albarelli, and Andrea Gasparetto. 2024. Hierarchical text classification and its foundations: A review of current research. *Electronics*, 13(7):1199.

Yu Zhang, Bowen Jin, Xiusi Chen, Yanzhen Shen, Yunyi Zhang, Yu Meng, and Jiawei Han. 2023. Weakly supervised multi-label classification of full-text scientific papers. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Yunyi Zhang, Ruozhen Yang, Xueqiang Xu, Rui Li, Jinfeng Xiao, Jiaming Shen, and Jiawei Han. 2025. Teleclass: Taxonomy enrichment and llm-enhanced hierarchical text classification with minimal supervision. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, pages 2032–2042.

A Evaluation Metrics

Following previous studies (Shen et al., 2021), we evaluate the multi-label classification results using Example-F1, Precision at k (P@k), and Mean Reciprocal Rank (MRR). Consistent with the notation in Section 3.2, let N be the total number of documents and $\mathcal{C}$ be the label set. For each document d_i, we denote its ground-truth label set as $Y_i^{true} \subset \mathcal{C}$ and the predicted label set as $Y_i^{pred} \subset \mathcal{C}$.

Macro-F1 calculates the average F1 score across all classes:

$$\text{Macro-F1} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{2 \cdot P_c \cdot R_c}{P_c + R_c} \tag{11}$$

where P_c and R_c denote the precision and recall for class $c \in \mathcal{C}$, respectively.

Example-F1 calculates the average F1 scores for all documents as follows:

$$\text{Example-F1} = \frac{1}{N} \sum_{i=1}^{N} \frac{2|Y_i^{true} \cap Y_i^{pred}|}{|Y_i^{true}| + |Y_i^{pred}|} \tag{12}$$

where Y_i^{true} and Y_i^{pred} denote the set of true and predicted labels for document d_i, respectively.

Precision at k (P@k) measures the proportion of correct predictions in the top-k predicted labels:

$$\text{P@}k = \frac{1}{N} \sum_{i=1}^{N} \frac{|Y_i^{true} \cap Y_{i,1:k}^{pred}|}{\min(k, |Y_i^{true}|)} \tag{13}$$

where $Y_{i,1:k}^{pred}$ represents the top-k most likely labels predicted by the model for document d_i. We report P@1 and P@3 in our experiments.

Mean Reciprocal Rank (MRR) evaluates the ranking quality of the predicted labels:

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|Y_i^{true}|} \sum_{c \in Y_i^{true}} \frac{1}{r_{ic}} \tag{14}$$

where r_{ic} is the rank of the true label c in the model's predicted list for document d_i.

B Prompt Templates

B.1 Prompt for Taxonomy Enrichment

Prompt template for taxonomy enrichment on the Amazon-531 dataset as follows.

> You are ChatGPT, a helpful AI chatbot specialized in classifying and analyzing Amazon product reviews. Your task is to understand the context of product reviews, categorize them into appropriate topics (e.g., quality, delivery, customer service), and assist in extracting useful insights from the reviews. The Amazon product tags c_j^{L-1} can be classified into more specific subcategories such as c_j^L. Please provide the subcategories in Python dictionary format, where the keys are the broader categories: c_j^L and the values are lists of 10 more specific subcategories.

Prompt template for taxonomy enrichment on the DBPedia-298 dataset as follows.

> You are ChatGPT, a helpful AI chatbot specialized in classifying and organizing Wikipedia article categories. Your task is to understand the context of these categories. You will categorize them into appropriate sub-topics OR provide specific, related instances if sub-topics are not suitable (e.g., for 'National Football League Season', instances like 'AFC' and 'NFC' are appropriate). Assist in creating a detailed knowledge hierarchy. The Wikipedia categories c_j^{L-1} can be classified into more specific subcategories such as c_j^L. Please provide the subcategories in Python dictionary format, where the keys are the broader categories: c_j^L and the values are lists of 10 more specific subcategories.

B.2 Prompt for Path Refinement

Prompt template for Path Refinement on the Amazon-531 dataset as follows.

> You are a highly precise, single-function product classification engine. Your SOLE task is to act as a strict path selector. You must output EXACTLY one string from the provided OPTIONS list, and nothing else. You are strictly forbidden from providing any text, explanation, quotation marks, or punctuation besides the chosen path itself.
> **CLASSIFICATION TASK**
> - **AMAZON PRODUCT REVIEW**: [Document Text]
> - **OPTIONS**: [P_{gen}]
> **INSTRUCTIONS**
> 1. Analyze the AMAZON PRODUCT REVIEW.
> 2. Select the single MOST RELEVANT category path from the OPTIONS.
> 3. Your FINAL OUTPUT MUST BE ONLY the selected path string. Do not include the number (e.g., "1.") or any other text.

Prompt template for Path Refinement on the DBPedia-298 dataset as follows.

> You are a highly precise classification engine for general knowledge and ontology. Your SOLE task is to act as a strict path selector. You must output EXACTLY one string from the provided OPTIONS list, and nothing else. You are strictly forbidden from providing any text, explanation, quotation marks, or punctuation besides the chosen path itself.
> **CLASSIFICATION TASK**
> - **WIKIPEDIA ARTICLE ABSTRACT**: [Document Text]
> - **OPTIONS**: [P_{gen}]
> **INSTRUCTIONS**
> 1. Analyze the WIKIPEDIA ARTICLE ABSTRACT.
> 2. Select the single MOST RELEVANT category path from the OPTIONS.
> 3. Your FINAL OUTPUT MUST BE ONLY the selected path string. Do not include the number (e.g., "1.") or any other text.