Automatic Construction of Error Taxonomies with Large Language Models

Anonymous ACL submission

Abstract

Error taxonomies organize error instances into structured categories and provide an important foundation for error analysis. However, most existing taxonomies are manually designed by experts, making them difficult to adapt to new domains and data distributions. In this paper, we propose an LLM-based framework for automatically constructing error taxonomies directly from raw error instances. The framework generates semantic error type labels with a large language model, builds hybrid representations that combine label semantics with transformation signals between erroneous and corrected expressions, and induces taxonomy structures through hierarchical clustering, followed by LLM-based semantic naming and structure refinement. Experiments on both grammatical errors from learner corpora and mathematical reasoning errors generated by large language models show the automatically constructed taxonomies achieve strong performance across multiple evaluation metrics, including exclusivity, coverage, consistency, and usability, and achieve competitive performance compared with manually designed taxonomies, outperforming them in several settings.

1 Introduction

Error analysis plays an important role in many natural language processing and educational applications, including grammatical error correction, automated essay scoring and intelligent tutoring systems (Dulay and Burt, 1972; Dodigovic, 2007; Heift and Schulze, 2007; Ye et al., 2022, 2023a,b; Huang et al., 2023; Zhang et al., 2023a; Li et al., 2023, 2025b,c). A common approach for understanding error patterns is through an *error taxonomy*, which organizes error instances into structured categories according to their underlying linguistic or reasoning mechanisms (Ma et al., 2022; Ye et al., 2025; Fei et al., 2023). Figure 1 illustrates a simple example of classifying an error instance into an abstract error type using an error taxonomy. Such taxonomies provide an essential bridge between individual error instances and higher-level explanations, enabling systematic error analysis, model evaluation, and targeted feedback generation (Corder, 2015; Bialystok et al., 1982).

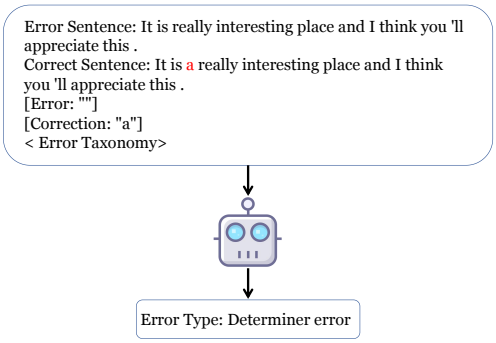

Figure 1: Example of mapping an error instance to an abstract error type using an error taxonomy.

Error taxonomies have been widely used across different domains. For example, grammatical error taxonomies are commonly adopted in learner corpus research to analyze language acquisition patterns (Ma et al., 2022; Ye et al., 2025; Fei et al., 2023), while reasoning error taxonomies have recently been used to study the failure modes of large language models in mathematical word problem solving and reasoning tasks (Sun et al., 2025; McNichols et al., 2023; Li et al., 2024). However, most existing error taxonomies are manually designed by experts. Although expert-designed taxonomies often provide valuable insights, they suffer from several practical limitations. First, manual construction is labor-intensive and difficult to scale when large datasets or multiple domains are involved. Second, category definitions often rely heavily on expert intuition, which may introduce subjective inconsistencies across datasets and research communities (Politzer and Ramirez, 1973; Tucker et al., 1974; Bryant et al., 2017). Third, as the distribution

of errors evolves with new models and tasks, manually maintaining and extending such taxonomies becomes increasingly challenging.

Recent advances in large language models (LLMs) have demonstrated strong capabilities in semantic abstraction and structure induction (Li et al., 2025a; Wang et al., 2023; Zeng et al., 2024), enabling new possibilities for automatically organizing textual information. However, automatically constructing error taxonomies remains challenging. Unlike traditional concept taxonomy construction, where hierarchical relations are defined over existing concepts, error taxonomy construction requires abstracting category definitions directly from diverse error instances. Furthermore, error data often contains paired information describing the relationship between erroneous outputs and their corrected counterparts, which provides important signals about the underlying causes of errors but is rarely utilized in existing approaches.

In this work, we propose an **LLM-based framework for the automatic construction of error taxonomies**. Our framework is primarily designed for constructing taxonomies of English grammatical errors, where large-scale learner data is available and fine-grained error categorization is essential for language learning and evaluation. Our approach first leverages a large language model to abstract concise error type labels from error instances. We then construct semantic representations that jointly encode the semantics of the generated error types and the transformation signals between erroneous and correct expressions. Based on these representations, an adaptive hierarchical clustering algorithm is used to construct the taxonomy structure, while an LLM-based naming module generates interpretable category labels for each cluster, followed by a structure optimization stage that refines the hierarchy and improves semantic consistency.

To evaluate the generalization ability of the framework, we further apply it to a different error analysis scenario, namely mathematical reasoning errors. Experimental results show that the automatically constructed taxonomies achieve strong performance across multiple evaluation dimensions. These results demonstrate that the proposed approach can effectively construct high-quality error taxonomies across different domains, despite variations in the optimal taxonomy structure, with minimal human intervention.

Our contributions are summarized as follows[1]:

- We formulate the task of **automatic error taxonomy construction** and propose an LLM-based framework that derives hierarchical error taxonomies directly from error instances.
- We introduce an **error-oriented representation** that captures error semantics and error–correction transformation signals, enabling more reliable clustering of error patterns.
- We conduct experiments on **English grammatical** and **mathematical reasoning errors**, demonstrating that the proposed framework constructs high-quality taxonomies competitive with expert-designed ones.

2 Related Work

2.1 Taxonomy Induction

Automatic taxonomy induction has long been studied for knowledge organization and semantic structure learning. Early work mainly relied on lexical–syntactic patterns and graph-based methods to extract hypernym–hyponym relations from large text corpora (Kozareva and Hovy, 2010). Later studies modeled hierarchical relations in continuous embedding spaces, for example by learning transformations between hyponym and hypernym embeddings (Fu et al., 2014), or by formulating taxonomy construction as a global structure optimization problem such as reinforcement learning based taxonomy generation (Mao et al., 2018).

Recent work has begun to explore the use of large language models for taxonomy construction. Some approaches use language models to predict candidate hierarchical relations before building taxonomy structures under structural constraints (Chen et al., 2023). Others directly generate taxonomy structures with LLMs. For example, TaxonomyGPT formulates taxonomy construction as a conditional text generation task (Li et al., 2025a), while Chain-of-Layer incrementally expands taxonomy structures with layer-wise generation and validation mechanisms (Zeng et al., 2024).

However, most existing studies focus on organizing entities or topical concepts. Taxonomy induction for *error classification*, where categories must be abstracted directly from error instances and may involve error–correction signals, remains relatively underexplored in the literature.

[1]All codes and data will be released after the review.

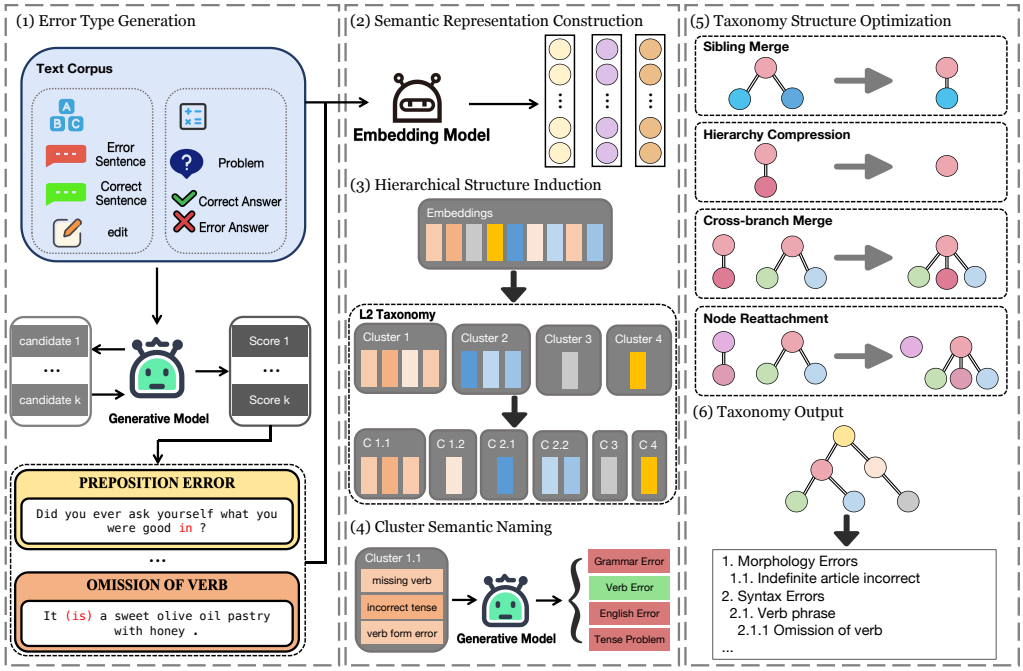

Figure 2: Overview of the proposed LLM-based error taxonomy construction framework

2.2 LLM-based Text Clustering

Text clustering is a widely used approach for discovering latent semantic structures from unlabeled data and is often adopted as a basis for taxonomy construction. Recent work incorporates LLMs into clustering in several complementary ways. One line of research improves text representations using high-quality LLM-generated embeddings before applying traditional clustering algorithms (Petukhova et al., 2025). CLUSTERLLM further leverages LLMs to generate semantic triplet constraints that guide contrastive representation learning for clustering (Zhang et al., 2023b). Another line of work performs clustering directly with LLMs. For example, LLM-MemCluster treats clustering as an iterative labeling process with dynamically maintained cluster labels (Zhu et al., 2025), while other approaches adopt a two-stage framework where LLMs first generate cluster labels and then assign texts to them accordingly (Huang and He, 2025). LLMs have also been used to refine clustering structures, such as boundary-aware reassignment with LLM reasoning (Feng et al., 2024), integrating cluster-level semantic descriptions (Pattnaik et al., 2024), or guiding clustering with natural language objectives(Wang et al., 2023).

Despite these advances, most work focuses on topic discovery or semantic similarity grouping, while clustering *error instances* for structured error taxonomy construction remains underexplored.

3 Method

We propose an LLM-based framework for automatically constructing error taxonomies from raw error instances. As illustrated in Figure 2, the framework first generates concise error type labels using an LLM, and then constructs semantic representations that integrate label semantics with transformation signals between erroneous and corrected expressions. Hierarchical clustering is subsequently applied to induce the taxonomy structure, followed by LLM-based cluster naming and a structure optimization stage that removes redundant nodes and improves overall hierarchical consistency.

3.1 Error Type Generation

To construct the taxonomy, we first generate an error type label for each error instance. Given an error instance represented as a pair consisting of an erroneous expression and its corresponding correction, the goal is to summarize the underlying error pattern into a concise semantic label.

Leveraging the semantic reasoning ability of large language models, we generate candidate error type labels for each instance. To improve robustness, we adopt a multi-candidate generation strategy with LLM-based self-evaluation, enabling more reliable label selection. For each error instance x, multiple candidate labels are generated:

$$T(x) = \{t_1, t_2, \ldots, t_k\} \quad (1)$$

Each candidate label is evaluated according to four criteria: **Alignment**, **Specificity**, **Reusability**, and **Consistency**. Alignment measures whether the label accurately captures the semantic difference between the erroneous expression and its correction, reflecting the underlying cause of the error. Specificity evaluates whether the label has an appropriate level of abstraction, avoiding overly generic or overly specific descriptions. Reusability measures whether the label can serve as a stable category name that can be reused across multiple error instances. Consistency evaluates whether the label appears consistently across multiple generations, indicating higher semantic stability. The overall quality score of a candidate label is defined as

$$q(t) = s_{\text{align}} + s_{\text{spec}} + s_{\text{reuse}} + s_{\text{cons}} \tag{2}$$

The final error type label is selected as

$$t^* = \arg\max_{t \in T(x)} q(t) \tag{3}$$

3.2 Semantic Representation Construction

After generating error type labels, we construct semantic representations for error instances to support similarity-based taxonomy induction. The representation encodes two complementary signals: the semantic meaning of error type labels and the transformation signals between erroneous and corrected expressions. First, the generated error type label is embedded into a semantic vector

$$\mathbf{e}_{type} = \text{Embedding}(t^*) \tag{4}$$

Next, we encode both the erroneous expression and the corrected expression and compute their semantic difference

$$\mathbf{e}_{diff} = \text{Embedding}(x_{correct}) - \text{Embedding}(x_{error}) \tag{5}$$

which captures the semantic transformation during error correction. The final representation is obtained by concatenating these two vectors

$$\mathbf{e} = \text{concat}(\mathbf{e}_{type}, \mathbf{e}_{diff}) \tag{6}$$

This representation jointly captures abstract error semantics and correction transformations, enabling more accurate and robust measurement of similarity between error instances.

3.3 Hierarchical Structure Induction

Given the semantic representations of error instances, we construct the taxonomy using an adaptive hierarchical clustering procedure. Starting from a root cluster containing all instances, clusters are recursively partitioned to form a hierarchical taxonomy with progressively refined structure. For a cluster C containing n samples, the number of subclusters is determined adaptively as

$$k = \min(K, \max(2, \lfloor n/3 \rfloor)) \tag{7}$$

where K denotes the maximum number of clusters allowed at each split. Cluster compactness is measured using the sum of squared errors (SSE)

$$SSE(C) = \sum_{x \in C} \|x - \mu_C\|^2 \tag{8}$$

where μ_C is the centroid of cluster C. After splitting C into k subclusters $C_1, \ldots, C_k$, the total within-cluster SSE becomes

$$SSE_{split} = \sum_{i=1}^{k} SSE(C_i) \tag{9}$$

The improvement ratio is defined as

$$\Delta = \frac{SSE(C) - SSE_{split}}{SSE(C)} \tag{10}$$

If $\Delta < \tau$, the cluster is considered sufficiently coherent and the splitting process stops; otherwise the resulting subclusters are added as child nodes in the taxonomy tree. The procedure continues recursively until the maximum depth is reached or clusters become stable. The detailed clustering algorithm is provided in Appendix A.

3.4 Cluster Semantic Naming

After clustering, each cluster contains a set of semantically similar error types but lacks an interpretable category name. To transform the clustering structure into a readable taxonomy, we use an LLM to generate candidate semantic labels for each cluster. For a cluster C, the set of error type labels within the cluster is provided as input to the LLM, which produces a set of candidate category names

$$L(C) = \{l_1, l_2, \ldots, l_m\} \tag{11}$$

To select the most appropriate label, we evaluate each candidate using two criteria: *coverage* and

confusion. Coverage measures how well a label represents the semantics of the current cluster

$$Cov(l, C) = \frac{1}{|C|} \sum_{t \in C} sim(l, t) \tag{12}$$

where $sim(\cdot)$ denotes semantic similarity. Confusion measures how well the label also describes other clusters at the same level

$$Conf(l, C) = \frac{1}{|S(C)|} \sum_{C' \in S(C)} Cov(l, C') \tag{13}$$

where $S(C)$ denotes the set of sibling clusters of C. The final score is defined as

$$Score(l, C) = Cov(l, C) - Conf(l, C) \tag{14}$$

and the final cluster label is selected as

$$l^* = \arg\max_{l \in L(C)} Score(l, C) \tag{15}$$

This procedure assigns interpretable and representative category names to clusters while preserving clear semantic boundaries and minimizing overlap between sibling clusters.

3.5 Taxonomy Structure Optimization

The taxonomy obtained from hierarchical clustering may still contain redundant nodes or structurally inconsistent relations, since the hierarchy is constructed primarily from local semantic similarity. To further improve the structural quality of the taxonomy, we introduce a structure optimization stage that iteratively refines the hierarchy. We consider a set of candidate structural operations that modify the taxonomy while preserving its hierarchical organization, including sibling merging, hierarchy compression, cross-branch merging, and node reattachment. Details of these operations are provided in Appendix B.

To evaluate the quality of a taxonomy T, we define the following objective function

$$\begin{aligned} J(T) = & \frac{1}{|E(T)|} \sum_{(p,c) \in E(T)} d(p, c) \\ & + \frac{1}{|Sib(T)|} \sum_{(u,v) \in Sib(T)} \max(0, \tau_s - d(u, v)) \\ & + \frac{|T|}{|T_0|} \end{aligned} \tag{16}$$

The three terms respectively encourage parent–child semantic coherence, sibling separability, and structural compactness, jointly defining a balanced objective for hierarchy refinement. The distance function is defined as

$$d(u, v) = \frac{1 - \cos(u, v)}{2} \tag{17}$$

where $E(T)$ denotes the set of parent–child edges and $Sib(T)$ denotes the set of sibling node pairs. During optimization, a candidate operation producing taxonomy T' is accepted if it reduces the objective value

$$\Delta J = J(T') - J(T) < 0 \tag{18}$$

This greedy procedure is applied iteratively until no further improvement is achieved, yielding a more compact and semantically coherent taxonomy across the hierarchy.

3.6 Taxonomy Output

Finally, the optimized taxonomy is converted into a standardized hierarchical representation. Each node is assigned a hierarchical index through depth-first traversal (e.g., 1, 1.1, 1.1.1), which preserves the parent–child relations within the taxonomy. This representation facilitates downstream applications such as automatic error classification, error analysis, and educational feedback generation.

4 Experiments

4.1 Experimental Setup

To evaluate the effectiveness of the proposed automatic error taxonomy construction method, experiments are conducted on two types of error datasets: grammatical errors from learner corpora and mathematical reasoning errors generated by large language models. These datasets provide diverse error instances from different domains, enabling us to construct error taxonomies and evaluate their structural quality using the proposed evaluation metrics.

Datasets. We conduct experiments on two datasets: a grammatical error dataset from learner corpora and a mathematical reasoning error dataset.

For grammatical errors, we use the dataset introduced in (Zou et al., 2025), derived from the W&I+LOCNESS corpus (Bryant et al., 2019). It contains 487 single-error instances annotated under four representative grammatical error taxonomies: POL73 (Politzer and Ramirez, 1973), TUC74 (Tucker et al., 1974), BRY17 (Bryant et al., 2017), and FEI23 (Fei et al., 2023), which serve as reference taxonomies in our evaluation.

Taxonomy	Exclusivity ↑	Coverage ↑	Consistency (ITC / ITS)↑	Usability (Macro / Micro)↑
(A) Existing Error Taxonomies				
POL73	0.752	0.698	0.215 / 0.894	0.562 / 0.534
TUC74	0.250	0.160	0.158 / **0.950**	0.315 / 0.119
BRY17	**0.928**	0.980	0.288 / 0.931	0.626 / **0.821**
FEI23	0.853	0.924	0.221 / 0.824	0.645 / 0.772
(B1) Proposed Method				
Flat KMeans (L1K50)	0.784	0.971	0.270 / 0.895	0.580 / 0.737
Hierarchical KMeans (L2K15)	0.820	0.985	0.310 / 0.937	0.565 / 0.719
Hierarchical KMeans (L3K6)	0.875	0.995	**0.318** / 0.945	**0.710** / 0.820
Hierarchical KMeans (L4K4)	0.645	0.688	0.221 / 0.895	0.547 / 0.511
(B2) Module Ablation				
L3K6 (Single Generation)	0.842	0.972	0.311 / 0.938	0.689 / 0.804
L3K6 (Only Type)	0.802	0.951	0.305 / 0.930	0.656 / 0.760
L3K6 (Only Diff)	0.550	0.480	0.222 / 0.887	0.595 / 0.721
L3K6 (Direct Naming)	0.868	0.993	0.316 / 0.942	0.701 / 0.815
L3K6 (w/o Structural Refinement)	0.740	**0.998**	0.300 / 0.935	0.633 / 0.721
(B3) Backbone Variants (Embedding / LLM)				
L3K6 + KALM Embedding	0.847	0.992	0.289 / 0.930	0.686 / 0.797
L3K6 + LLaMA Embedding	0.855	0.978	0.307 / 0.936	0.691 / 0.780
L3K6 + Gemini Generation	0.880	0.983	0.314 / **0.950**	0.703 / 0.803
L3K6 + Claude Generation	0.877	0.997	0.316 / 0.941	0.695 / 0.799

Table 1: Performance comparison of English error taxonomies. All results are obtained using GPT-5-mini as the evaluator. LxKy denotes a hierarchical clustering configuration with x levels and at most y sub-clusters per parent node. **Bold** indicates the best result and underline indicates the second-best.

For mathematical reasoning errors, we construct an experimental dataset based on MWPES-300k (Sun et al., 2025), which consists of incorrect reasoning outputs generated by large language models on several math word problem benchmarks, including SVAMP (Patel et al., 2021), GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), and AQuA (Ling et al., 2017). Each instance contains a math problem, a model-generated incorrect solution, and the corresponding ground-truth answer.

To keep the experiments computationally manageable, we randomly sample 500 erroneous instances from this dataset and annotate them following the procedure described in (Zou et al., 2025).

Models. Two types of models are used in the taxonomy construction framework: embedding models for semantic representation and generative models for error type and cluster name generation.

For semantic representation, we primarily use Qwen3-Embedding-8B (Zhang et al., 2025). To analyze the impact of embedding models on taxonomy construction, two additional embedding models are included in ablation experiments: KaLM-Embedding (Hu et al., 2025) and LLaMA-Embed-Nemotron-8B (Babakhin et al., 2025).

For error type generation and cluster semantic naming, we primarily use GPT-5-mini (OpenAI, 2024). To evaluate the robustness of the framework across different generative models, we additionally test Claude-Haiku-4.5 (Anthropic, 2025) and Gemini-3-Flash-preview (Google DeepMind, 2025). All generation experiments use a unified prompting template and identical decoding parameters to ensure comparability.

Evaluation Metrics. To evaluate the quality of the constructed error taxonomies, we adopt the taxonomy evaluation framework proposed in (Zou et al., 2025), which includes four dimensions: exclusivity, coverage, consistency, and usability.

Exclusivity evaluates whether category boundaries are clearly defined, coverage measures how well the taxonomy captures observed error instances, consistency assesses structural stability in semantic space through intra-type consistency (ITC) and inter-type separability (ITS), and usability reflects the practical applicability of the taxonomy for automatic classification and human annotation. Detailed definitions of these metrics are provided in Appendix C.

4.2 Main Results and Analysis

The main experimental results are presented and summarized in Table 1.

Comparison with Existing Taxonomies. Compared with manually designed taxonomies, the au-

Taxonomy	Exclusivity ↑	Coverage ↑	Consistency (ITC / ITS)↑	Usability (Macro / Micro)↑
(A) Original Taxonomy				
Original	0.572	0.618	0.765 / 0.250	0.206 / 0.364
(B) Ours				
Flat KMeans (L1K50)	**0.647**	**0.992**	**0.793** / 0.192	**0.540 / 0.572**
Hierarchical KMeans (L2K15)	0.541	0.880	0.724 / 0.320	0.144 / 0.320
Hierarchical KMeans (L3K6)	0.473	0.764	0.730 / 0.318	0.128 / 0.336
Hierarchical KMeans (L4K4)	0.193	0.526	0.733 / **0.323**	0.110 / 0.180

Table 2: Performance comparison of Math error taxonomies. All results are obtained using GPT-5-mini as the evaluator. **Bold** indicates the best result and underline indicates the second-best.

Setting	Exclusivity	Coverage	Consistency (ITC / ITS)	Usability (Macro / Micro)
Flat KMeans (L1K50)	0.783 ± 0.006	0.970 ± 0.004	0.269 ± 0.005 / 0.894 ± 0.006	0.579 ± 0.009 / 0.736 ± 0.010
Hierarchical KMeans (L2K15)	0.819 ± 0.008	0.984 ± 0.006	0.309 ± 0.007 / 0.936 ± 0.005	0.563 ± 0.010 / 0.717 ± 0.011
Hierarchical KMeans (L3K6)	0.874 ± 0.006	0.994 ± 0.003	0.317 ± 0.005 / 0.944 ± 0.004	0.708 ± 0.008 / 0.819 ± 0.009
Hierarchical KMeans (L4K4)	0.642 ± 0.015	0.686 ± 0.017	0.220 ± 0.012 / 0.893 ± 0.010	0.545 ± 0.018 / 0.509 ± 0.020

Table 3: Robustness analysis under different random seeds (mean $\pm$ standard deviation).

tomatically constructed taxonomy achieves more balanced performance across different metrics. In particular, the L3K6 configuration achieves the highest intra-type consistency (ITC = 0.318) and the best Macro F1 score (0.710), while maintaining strong performance on other metrics. It also achieves higher coverage (0.995) than the best manual taxonomy BRY17 (0.980), indicating that data-driven semantic grouping can capture a broader range of error patterns. Although BRY17 achieves the highest exclusivity and Micro F1, its performance is less balanced across different metrics.

Effect of Hierarchical Structure. Further analysis shows that hierarchical structure significantly influences taxonomy quality. Flat clustering (L1K50) achieves high coverage but performs worse in exclusivity and usability, suggesting that purely flat structures fail to capture meaningful semantic organization. Introducing hierarchical structures improves consistency and usability. Among all configurations, the three-level structure (L3K6) provides the best balance between semantic granularity and structural stability. However, increasing the hierarchy depth further (L4K4) leads to performance degradation across most metrics, indicating that excessive hierarchy depth may cause over-fragmentation of categories.

Module Ablation. Ablation experiments further reveal the contributions of different components. Removing multi-candidate generation (Single Generation) slightly reduces performance, demonstrating the importance of generation diversity and self-evaluation filtering. Using only semantic error type labels (Only Type) leads to moderate degradation, while relying only on difference embeddings (Only Diff) causes a substantial performance drop across metrics. This indicates that semantic error labels provide the primary information, while transformation signals serve as complementary cues. Removing structural refinement slightly increases coverage but decreases exclusivity and consistency, suggesting that structure optimization helps eliminate redundant nodes and clarify category boundaries.

Backbone Robustness. Replacing the embedding model or generation model only causes minor performance changes. Across all backbone variants, the L3K6 structure consistently achieves strong results, indicating that the effectiveness of the proposed framework mainly comes from the overall taxonomy construction pipeline rather than relying on a specific model.

4.3 Mathematical Error Taxonomy Evaluation

To further evaluate the generalization ability of the proposed method, experiments are conducted on the mathematical error dataset. The results are shown in Table 2 for detailed comparison.

Compared with grammatical error analysis, the mathematical task exhibits a different performance pattern. Flat KMeans achieves the best overall results across most metrics, including coverage (0.992), exclusivity (0.647), intra-type consistency

(ITC = 0.793), and usability (0.540 / 0.572). These results indicate that flat clustering is more suitable for organizing mathematical reasoning errors. This difference arises from the nature of reasoning errors in mathematical problem solving. Such errors are typically local computation or reasoning mistakes rather than hierarchical conceptual relations. As a result, introducing deeper hierarchical structures tends to fragment semantically related errors and leads to lower exclusivity and usability.

Although Flat KMeans does not utilize the full hierarchical structure of our framework, the automatically constructed taxonomies still significantly outperform the manually designed taxonomy across most metrics. In particular, the coverage improves from 0.618 to 0.992 and usability increases substantially. This demonstrates that the main improvements come from the semantic representation learning and error type generation stages in the proposed pipeline, which enable the taxonomy to better capture the distribution of reasoning error patterns than manually designed error taxonomy.

4.4 Robustness to Random Initialization

Since the hierarchical clustering stage uses KMeans, different random seeds may affect the results. To evaluate robustness, we repeat the taxonomy construction process five times with different seeds and report the mean and standard deviation of the evaluation metrics in Table 3.

Overall, the variance across different seeds is small, indicating that the proposed method is robust to initialization randomness. In particular, the best-performing configuration L3K6 shows very stable results with minimal standard deviations across all metrics. Although deeper hierarchies such as L4K4 exhibit slightly larger fluctuations, the variations remain within a reasonable range.

These results demonstrate that the performance of the proposed taxonomy construction framework mainly arises from the overall pipeline design rather than specific initialization conditions.

4.5 Human Annotation Agreement

To further evaluate the usability of automatically constructed taxonomies, we measure inter-annotator agreement using Cohen's Kappa (κ) (Cohen, 1960), which quantifies agreement beyond chance:

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \tag{19}$$

Taxonomy	A1 & A2	A2 & A3	A1 & A3	Average
(A) Existing Error Taxonomies				
POL73	0.663	0.619	0.632	0.638
TUC74	0.641	0.531	0.557	0.576
BRY17	0.817	0.767	0.803	0.796
FEI23	0.703	0.770	0.711	0.728
(B) Proposed Method				
Flat KMeans (L1K50)	0.777	0.749	0.733	0.753
Hierarchical KMeans (L2K15)	0.783	0.737	0.748	0.756
Hierarchical KMeans (L3K6)	**0.861**	**0.795**	**0.820**	**0.825**
Hierarchical KMeans (L4K4)	0.703	0.695	0.711	0.703

Table 4: Inter-annotator agreement (Cohen's κ) for English error taxonomies.

where p_o denotes the observed agreement between annotators and p_e denotes the expected agreement estimated from marginal label distributions across all categories.

As shown in Table 4, the automatically constructed taxonomy L3K6 achieves the highest agreement score (0.825), outperforming both other automatic structures and manually designed taxonomies. This indicates that the proposed taxonomy provides clearer semantic boundaries and more interpretable category definitions. Flat KMeans and L2K15 show slightly lower agreement, while deeper hierarchies such as L4K4 yield lower agreement due to increased structural complexity. These results are consistent with the usability metrics observed in the main experiments.

5 Conclusion

We present an LLM-based framework for automatically constructing error taxonomies from raw error instances. Our method generates semantic error type labels using a large language model, builds hybrid representations that integrate label semantics with error–correction transformation signals, and induces taxonomy structures through hierarchical clustering followed by semantic naming and structure refinement. Experiments on grammatical errors and mathematical reasoning errors show that the automatically constructed taxonomies achieve strong performance across multiple evaluation metrics and remain competitive with manually designed taxonomies. These results demonstrate that large language models can effectively support the automatic construction of structured error taxonomies across different domains with minimal human intervention.

Limitations

Although the proposed framework shows strong performance in automatically constructing error taxonomies, several limitations remain.

First, the hierarchical clustering stage still requires predefined structural parameters, such as the maximum hierarchy depth and the upper bound on the number of clusters per split. While the adaptive splitting strategy helps mitigate this issue, the overall taxonomy structure still depends on these preset parameters. In future work, it would be interesting to explore fully adaptive approaches where both the hierarchy depth and the number of clusters can be determined automatically from the data.

Second, the current framework is primarily designed and evaluated on English grammatical errors. When applied to other domains, such as mathematical reasoning errors, certain modules in the pipeline become less effective due to differences in the structural properties of error patterns. Although the overall pipeline can still produce reasonable taxonomies in such scenarios, developing a more domain-general taxonomy construction framework remains an important direction for future research.

Ethical Considerations

We conduct our experiments using publicly available datasets, including learner corpus data derived from W&I+LOCNESS and mathematical reasoning error data constructed from MWPES-300k. These datasets do not contain personally identifiable or sensitive information, and we use them solely for research purposes in accordance with their original licenses and intended usage.

We use publicly available language models and embedding models in our experiments and properly acknowledge their sources. All models are applied following their recommended usage guidelines.

For the annotation-related parts of this study, we recruited three postgraduate students with backgrounds in linguistics and related fields as annotators. We compensated them at a standard hourly rate and ensured that the workload remained reasonable. The annotation process involved routine research tasks and did not expose annotators to sensitive or harmful content.

Despite these considerations, potential risks remain. LLM-based label generation may introduce biases or inconsistencies, and the resulting taxonomies may be over-interpreted in downstream applications. Our framework mitigates these risks through repeated label generation, hybrid representations, and structure refinement, which improve consistency and robustness. Nevertheless, the constructed taxonomies should be treated as supportive analytical tools rather than definitive standards.

References

Anthropic. 2025. Introducing claude haiku 4.5. https://www.anthropic.com/news/claude-haiku-4-5. Published: Oct. 15, 2025.

Yauhen Babakhin, Radek Osmulski, Ronay Ak, Gabriel Moreira, Mengyao Xu, Benedikt Schifferer, Bo Liu, and Even Oldridge. 2025. Llama-embed-nemotron-8b: A universal text embedding model for multilingual and cross-lingual tasks. *Preprint*, arXiv:2511.07025.

Ellen Bialystok, Heidi Dulay, Marina Burt, and Stephen Krashen. 1982. Language two. *The Modern Language Journal*, 67:273.

Christopher Bryant, Mariano Felice, Øistein E Andersen, and Ted Briscoe. 2019. The bea-2019 shared task on grammatical error correction. In *Proceedings of the fourteenth workshop on innovative use of NLP for building educational applications*, pages 52–75.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th annual meeting of the association for computational linguistics (Volume 1: Long Papers)*, pages 793–805.

Boqi Chen, Fandi Yi, and Dániel Varró. 2023. Prompting or fine-tuning? a comparative study of large language models for taxonomy construction. In *2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, page 588–596. IEEE Press.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Stephen Pit Corder. 2015. The significance of learners' errors. In *Error analysis*, pages 19–27. Routledge.

Marina Dodigovic. 2007. Artificial intelligence and second language learning: An efficient approach to error remediation. *Language Awareness*, 16(2):99–113.

Heidi C Dulay and Marina K Burt. 1972. Goofing: An indicator of children's second language learning strategies 1. *Language learning*, 22(2):235–252.

Yuejiao Fei, Leyang Cui, Sen Yang, and 1 others. 2023. Enhancing grammatical error correction systems with explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 7489–7501.

Zijin Feng, Luyang Lin, Lingzhi Wang, Hong Cheng, and Kam-Fai Wong. 2024. LLMEdgeRefine: Enhancing text clustering with LLM-based boundary point refinement. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18455–18462, Miami, Florida, USA. Association for Computational Linguistics.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1199–1209, Baltimore, Maryland. Association for Computational Linguistics.

Google DeepMind. 2025. Best for frontier intelligence at speed. `https://deepmind.google/models/gemini/flash/`.

Trude Heift and Mathias Schulze. 2007. *Errors and Intelligence in Computer-Assisted Language Learning: Parsers and Pedagogues*. Routledge.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset.

Xinshuo Hu, Zifei Shan, Xinping Zhao, Zetian Sun, Zhenyu Liu, Dongfang Li, Shaolin Ye, Xinyuan Wei, Qian Chen, Baotian Hu, Haofen Wang, Jun Yu, and Min Zhang. 2025. Kalm-embedding: Superior training data brings a stronger embedding model. *Preprint*, arXiv:2501.01028.

Chen Huang and Guoxiu He. 2025. Text clustering as classification with llms. In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, SIGIR-AP 2025, page 374–384, New York, NY, USA. Association for Computing Machinery.

Haojing Huang, Jingheng Ye, Qingyu Zhou, Yinghui Li, Yangning Li, Feng Zhou, and Hai-Tao Zheng. 2023. A frustratingly easy plug-and-play detection-and-reasoning module for chinese spelling check. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11514–11525.

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.

Jiaye Li, Yuan Meng, Lijun Wang, Tianhao Qian, Songlin Zhai, and Guilin Qi. 2025a. Taxonomy induction using llms: An enhanced framework by integrating doubly-checked mechanism and self-evaluation strategy. In *China Conference on Knowledge Graph and Semantic Computing and International Joint Conference on Knowledge Graphs*, pages 134–146, Singapore. Springer Nature Singapore.

Xiaoyuan Li, Wenjie Wang, Moxin Li, Junrong Guo, Yang Zhang, and Fuli Feng. 2024. Evaluating mathematical reasoning of large language models: A focus on error identification and correction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11316–11360, Bangkok, Thailand. Association for Computational Linguistics.

Yinghui Li, Haojing Huang, Shirong Ma, Yong Jiang, Yangning Li, Feng Zhou, Hai-Tao Zheng, and Qingyu Zhou. 2023. On the (in) effectiveness of large language models for chinese text correction. *arXiv preprint arXiv:2307.09007*.

Yinghui Li, Shirong Ma, Shaoshen Chen, Haojing Huang, Shulin Huang, Yangning Li, Hai-Tao Zheng, and Ying Shen. 2025b. Correct like humans: Progressive learning framework for chinese text error correction. *Expert Systems with Applications*, 265:126039.

Yinghui Li, Shang Qin, Jingheng Ye, Haojing Huang, Yangning Li, Shu-Yu Guo, Libo Qin, Xuming Hu, Wenhao Jiang, Hai-Tao Zheng, and 1 others. 2025c. Rethinking the roles of large language models in chinese grammatical error correction. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 553–567.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Shirong Ma, Yinghui Li, Rongyi Sun, Qingyu Zhou, Shulin Huang, Ding Zhang, Li Yangning, Ruiyang Liu, Zhongli Li, Yunbo Cao, and 1 others. 2022. Linguistic rules-based corpus generation for native chinese grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 576–589.

Yuning Mao, Xiang Ren, Jiaming Shen, Xiaotao Gu, and Jiawei Han. 2018. End-to-end reinforcement learning for automatic taxonomy induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2462–2472, Melbourne, Australia. Association for Computational Linguistics.

Hunter McNichols, Mengxue Zhang, and Andrew Lan. 2023. Algebra error classification with large language models. In *Artificial Intelligence in Education: 24th International Conference, AIED 2023, Tokyo, Japan, July 3–7, 2023, Proceedings*, page 365–376, Berlin, Heidelberg. Springer-Verlag.

OpenAI. 2024. Introducing GPT-5. https://openai.com/index/introducing-gpt-5/. Accessed: Aug. 7, 2025.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2080–2094.

Anup Pattnaik, Cijo George, Rishabh Kumar Tripathi, Sasanka Vutla, and Jithendra Vepa. 2024. Improving hierarchical text clustering with LLM-guided multi-view cluster representation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 719–727, Miami, Florida, US. Association for Computational Linguistics.

Alina Petukhova, João P. Matos-Carvalho, and Nuno Fachada. 2025. Text clustering with large language model embeddings. *International Journal of Cognitive Computing in Engineering*, 6:100–108.

Robert L. Politzer and Arnulfo G. Ramirez. 1973. An error analysis of the spoken english of mexican-american pupils in a bilingual school and a monolingual school. *Language Learning*, 23(1):39–61.

Yuhong Sun, Zhangyue Yin, Xuanjing Huang, Xipeng Qiu, and Hui Zhao. 2025. Error classification of large language models on math word problems: A dynamically adaptive framework.

G. R. Tucker, M. K. Burt, and C. Kiparsky. 1974. The gooficon: A repair manual for english. *TESOL Quarterly*, 8(2):191.

Zihan Wang, Jingbo Shang, and Ruiqi Zhong. 2023. Goal-driven explainable clustering via language descriptions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10626–10649, Singapore. Association for Computational Linguistics.

Jingheng Ye, Yinghui Li, Yangning Li, and Hai-Tao Zheng. 2023a. Mixedit: Revisiting data augmentation and beyond for grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10161–10175.

Jingheng Ye, Yinghui Li, Shirong Ma, Rui Xie, Wei Wu, and Hai-Tao Zheng. 2022. Focus is what you need for chinese grammatical error correction. *arXiv preprint arXiv:2210.12692*.

Jingheng Ye, Yinghui Li, and Haitao Zheng. 2023b. System report for ccl23-eval task 7: Thu kelab (sz)-exploring data augmentation and denoising for chinese grammatical error correction. In *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 3: Evaluations)*, pages 262–270.

Jingheng Ye, Shang Qin, Yinghui Li, Xuxin Cheng, Libo Qin, Hai-Tao Zheng, Ying Shen, Peng Xing, Zishan Xu, Guo Cheng, and 1 others. 2025. Excgec: A benchmark for edit-wise explainable chinese grammatical error correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25678–25686.

Qingkai Zeng, Yuyang Bai, Zhaoxuan Tan, Shangbin Feng, Zhenwen Liang, Zhihan Zhang, and Meng Jiang. 2024. Chain-of-layer: Iteratively prompting large language models for taxonomy induction from limited examples. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, page 3093–3102, New York, NY, USA. Association for Computing Machinery.

Ding Zhang, Yinghui Li, Qingyu Zhou, Shirong Ma, Yangning Li, Yunbo Cao, and Hai-Tao Zheng. 2023a. Contextual similarity is more valuable than character similarity: An empirical study for chinese spell checking. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

Yuwei Zhang, Zihan Wang, and Jingbo Shang. 2023b. ClusterLLM: Large language models as a guide for text clustering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13903–13920, Singapore. Association for Computational Linguistics.

Yuanjie Zhu, Liangwei Yang, Ke Xu, Weizhi Zhang, Zihe Song, Jindong Wang, and Philip S. Yu. 2025. Llm-memcluster: Empowering large language models with dynamic memory for text clustering. *Preprint*, arXiv:2511.15424.

Deqing Zou, Jingheng Ye, Yulu Liu, Yu Wu, Zishan Xu, Yinghui Li, Hai-Tao Zheng, Bingxu An, Zhao Wei, and Yong Xu. 2025. Revisiting classification taxonomy for grammatical errors. *arXiv preprint arXiv:2502.11890*.

A Adaptive Hierarchical Clustering Algorithm

Algorithm 1 describes the adaptive hierarchical clustering procedure used to construct the taxonomy structure.

Algorithm 1 Adaptive Hierarchical Clustering for Error Taxonomy Construction

Require: Embeddings $\mathbf{X}$, maximum depth L, maximum clusters per split K, minimum cluster size $n_{\min}$, threshold τ
Ensure: Taxonomy tree $\mathcal{T}$

Initialize root cluster $C^{(0)} \leftarrow \mathbf{X}$ and taxonomy tree $\mathcal{T}$
for $\ell = 1$ to L **do**
  for each cluster C at level $\ell - 1$ **do**
    if $|C| < n_{\min}$ **then**
      mark C as leaf
      continue
    end if
    $k \leftarrow \min(K, \max(2, \lfloor |C|/3 \rfloor))$
    $\{C_1, \ldots, C_k\} \leftarrow \textsc{KMeansSplit}(C, k)$
    compute $\Delta = \frac{SSE(C) - SSE_{split}}{SSE(C)}$
    if $\Delta < \tau$ **then**
      mark C as leaf
      continue
    end if
    for $i = 1$ to k **do**
      add C_i as child of C in $\mathcal{T}$
    end for
  end for
end for
return $\mathcal{T}$

B Structural Optimization Operations

To refine the taxonomy structure, we apply several structural operations that modify the hierarchy while preserving the overall tree organization. These operations aim to remove redundant nodes and improve semantic consistency within the taxonomy.

- **Sibling Merge.** If two sibling nodes under the same parent exhibit high semantic similarity, they are merged into a single node. The merged node inherits the union of their subtrees, reducing redundant categories.
- **Hierarchy Compression.** If a parent node contains only one child and the two nodes are semantically similar, the intermediate hierarchy level is removed by absorbing the child into the parent node.
- **Cross-branch Merge.** If nodes located in different branches but at the same hierarchy level are semantically similar, they may be merged into a single node, and their subtrees are unified.
- **Node Reattachment.** If a node is semantically more similar to another potential parent than its current parent, it can be reattached to the more appropriate parent while preserving its subtree structure.

These operations are evaluated using the objective function described in Section 3.5, and are applied iteratively until no further improvement can be achieved.

C Detailed definitions of Evaluation Metrics

This section briefly describes the evaluation metrics used in our experiments. All metrics follow the taxonomy evaluation framework proposed in (Zou et al., 2025). Here we provide a high-level description of how each metric is computed.

Exclusivity Exclusivity measures whether the error types in a taxonomy are mutually exclusive. The metric analyzes the confidence scores produced by large language models when assigning error categories to instances. If multiple error types receive high confidence for the same instance, this indicates overlapping category boundaries and reduces the exclusivity score. The final score is obtained by aggregating overlap statistics across the dataset.

Coverage Coverage evaluates how well the taxonomy accounts for the observed error instances. It is defined as the proportion of instances that can be assigned to predefined error categories (excluding "Other" categories). Higher coverage indicates that the taxonomy captures a larger portion of the observed error space.

Consistency Consistency measures the structural coherence of the taxonomy in semantic space. Error instances are mapped into an embedding space, and two complementary properties are evaluated: intra-type consistency (ITC) and inter-type separability (ITS). ITC measures the compactness of instances belonging to the same error type, while ITS evaluates the separability between different error types.

Usability Usability reflects the practical applicability of a taxonomy in real-world error classification scenarios. It is evaluated from two perspectives: model-level usability and human-level usability. Model-level usability measures how effectively models can apply the taxonomy for automatic classification, while human-level usability evaluates the consistency of human annotations when applying the taxonomy.