Bus Inf Syst Eng
https://doi.org/10.1007/s12599-025-00982-y

RESEARCH PAPER

Semi-Automatic Hierarchical Taxonomy Creation from Existing Taxonomies with Large Language Models

Elham Motamedi · Inna Novalija · Luis Rei

Received: 25 January 2025 / Accepted: 17 November 2025

Abstract The development of taxonomies is critical for organizing knowledge in the field of Information Systems, particularly in hierarchical structures that align with human cognitive and navigational preferences. Traditional manual methods for creating taxonomies require substantial expert involvement, which is often impractical under resource constraints. Moreover, these methods are frequently reported without sufficient documentation or evaluation. Therefore, a semi-automatic method is proposed for refining overly detailed taxonomies, where the initial groups are iteratively consolidated and generalized, resulting in a more abstract and practical taxonomy. The method uses large language models (LLMs) to automate the process by leveraging their contextual understanding and generative capabilities. Particularly, LLMs are employed to: (i) refine the taxonomy's perspective, (ii) decide on merging groups with siblings based on semantic similarity, and (iii) propose representative labels for merged groups. The method uses expert input to validate and fine-tune the various steps of the process. The proposed method was then applied to develop a taxonomy of innovations, using the Cooperative Patent Classification schema, a widely employed classification system for patent documents, as a case study. Since the taxonomy was intended to assist in organizing and classifying patents, the results were compared to those of a manually created taxonomy based on classification performance. The classifier using the taxonomy generated by the proposed method performed comparably to the manually created taxonomy (i.e., Micro-F scores of 0.70 vs. 0.71 and Macro-F1 scores of 0.87 vs. 0.86). Moreover, both taxonomies exhibited similar structural features and groups. Beyond creating an innovation taxonomy and using it for patent classification, the method has broader implications for efficiently generating taxonomies across different domains, offering a transparent and replicable approach.

Keywords Conceptual modeling · Taxonomy design · Large language models · Patents

1 Introduction

The development of taxonomies plays a critical role in organizing and structuring knowledge in the field of IS (Pena et al. 2024). Taxonomies enable researchers and practitioners to categorize concepts, identify relationships, and build a common understanding of complex domains (Kundisch et al. 2022). Developing taxonomies involves identifying key dimensions and attributes that define a domain, grouping related elements, and ensuring consistency and clarity in the classification (FakhrHosseini et al. 2024).

A common approach to modeling interrelations among concepts in taxonomies is through a hierarchical structure, typically represented by a tree (Gomez and Moens 2014). This organization aligns with natural human navigational behavior and has been shown to be effective in information retrieval (Trattner et al. 2012). Hierarchical taxonomies present relationships among elements, progressing from broader categories at the highest level to more specific ones at the lowest level (i.e., from root level to leaf level in a tree structure). Depending on the domain and application, hierarchical taxonomies are designed with varying levels

Accepted after two revisions by the editors of the Special Issue.

E. Motamedi (✉) · I. Novalija · L. Rei
Jožef Stefan Institute, Ljubljana, Slovenia
e-mail: elham.motamedi@ijs.si

Published online: 07 January 2026

and classes at each level to serve as knowledge management tools that help target users organize objects and retrieve information more efficiently (Kang et al. 2024). However, excessive detail and numerous classes at each level can complicate tracking items within the hierarchy, making it impractical for classification and retrieval purposes (Härtinger and Clarke 2016).

Fine-grained classes in taxonomies are typically defined by domain experts based on subtle distinctions within specific groups. As the hierarchy deepens, it becomes increasingly difficult to identify distinctive features that reliably characterize objects at lower levels (Yang et al. 2018a). Several studies have acknowledged the challenges of fine-grained classification. For instance, Yang et al. (2018b) proposed an automatic multi-agent cooperation method to improve performance in such tasks. Kamateri et al. (2024) also highlighted the difficulty of handling taxonomies with extensive class labels. In these taxonomies, predictive accuracy suffers due to the many classes and semantic similarity among them. To address this, studies such as Bekamiri et al. (2024) have focused on higher levels of the taxonomy (e.g., subclasses with 663 labels) to achieve higher performance. However, we argue that this fixed-level approach may overlook domain-specific needs. Established fields may benefit from detailed groups, while emerging areas may be better served by broader groupings, suggesting that granularity should adapt to the scope of each domain.

Our proposed method uses data-dependent signals, particularly class sizes, to identify candidate groups for merging in the abstraction process, which aims to create a more abstract taxonomy. By *more abstract*, we refer to a structural property of the resulting taxonomy in which highly specific or narrow categories are merged into broader, higher-level groupings. These generalized groupings aim to reduce granularity. This abstraction process may lead to a taxonomy that is also more practical, that is, better suited to support particular tasks such as classification. In this context, we define practicality as the ability of the taxonomy to fulfill end users' needs, which may include its performance in classification tasks and its structural coherence. While the level of abstraction is a characteristic of the taxonomy's structure, practicality reflects its task-oriented effectiveness. Similar strategies have been explored in prior semi-automatic abstraction approaches; however, these typically require training a classifier at each node, which can be computationally intensive (Babbar et al. 2016). In contrast, our method uses class size as a signal to suggest candidate groups for merging, with the final merging decision made by an LLM that evaluates semantic similarity among the candidate groups given their contextual knowledge across various domains (Yang et al. 2024). Alternatively, the candidate groups identified by our method could be presented to domain experts, who would manually decide which groups to merge. While this would reduce the scope of expert effort compared to fully manual taxonomy development, it would still require substantial human input. By delegating the merge decision to an LLM, our method further reduces expert workload while maintaining flexibility for expert oversight. Moreover, manual refinement of taxonomies has often been reported with low transparency and limited evaluation (Kozareva and Hovy 2010), whereas our proposed method, which adapts an established taxonomy design approach, is designed to allow transparent documentation and evaluation.

The proposed method instructs LLMs to identify groups within the taxonomy that share a high degree of knowledge, enabling their merger into a more generalized taxonomy. Moreover, using the generative power of LLMs, they are instructed to suggest representative labels for the merged groups. Using LLMs shifts experts' efforts from building taxonomies from scratch to guiding LLMs through the various steps of the method to generate the desired taxonomy.

To validate our proposed method, we applied it to develop a taxonomy of innovations, using the detailed CPC taxonomy as a case study. The CPC is a widely adopted classification system for organizing patent documents in a hierarchical structure (Gomez and Moens 2014). Given that the case study aimed to organize and classify patent documents, we compared the classification results achieved using our method's taxonomy with those obtained using a manually created taxonomy from the literature (Motamedi et al. 2024a, b). Our results showed that the taxonomy generated by our method achieved comparable classification performance to the manually created taxonomy, while requiring substantially less expert input.

The research questions that the study addresses are the following:

- *RQ1* Relative to the original CPC taxonomy, to what extent does the proposed semi-automatic LLM-assisted method reduce maximum depth, mean root-to-leaf path length, and the number of categories per level, while preserving semantic coherence, measured as the ratio of average intra- to inter-cluster similarity of associated text inputs?
- *RQ2* Compared with a manually created taxonomy from the literature, how effectively does the refined taxonomy support downstream classification, as illustrated on CPC documents and measured by Macro-F1 and Micro-F1 scores?

Based on the defined research questions, we formulated the following hypotheses:

- *H1* Relative to the original CPC taxonomy, the proposed semi-automatic LLM-assisted method produces a taxonomy with reduced maximum depth, reduced mean root-to-leaf path length, and fewer categories per level, while preserving semantic coherence, measured by the ratio of average intra- to inter-cluster similarity of associated text inputs.
- *H2* The abstracted taxonomy generated using our proposed method achieves comparable classification performance to a manually created taxonomy in a downstream patent classification task, as measured by Macro-F1 and Micro-F1 scores.

This study makes three key contributions by proposing a semi-automatic method and demonstrating its application in creating an innovation taxonomy: (i) We propose a semi-automatic method for generating a more abstract hierarchical taxonomy from an existing overly detailed one, ensuring it remains transparent and evaluable, (ii) We validate our proposed method by applying it to create a taxonomy of *innovations*, using the CPC schema as the initial taxonomy, and (iii) We compare the created taxonomy of innovations with a manually created one based on the defined evaluation goals, particularly the classification of patents in our case study.

Key Concepts and Evaluation Scope: To enhance understanding of our semi-automatic taxonomy refinement method, we define several key concepts and terms used in this study and clarify the rationale behind the evaluation approach.

Abstraction refers to the process of transforming a taxonomy by merging highly specific or fine-grained categories into broader, higher-level groupings (Wilsens et al. 2024).

Granularity captures the number of detailed groups present in the taxonomy. High granularity can reduce the performance of automatic classification approaches in classifying associated items to the categories defined in the taxonomy.

Practicality in our study refers to the usefulness of a taxonomy in supporting its intended task. In our case study, practicality is operationalized as the taxonomy's effectiveness in classifying domain-specific objects (e.g., patent documents).

Evaluation Focus: Taxonomies serve purposes such as clarifying concept relationships and enabling shared understanding of complex domains. When developed from scratch, they are often assessed using metrics such as comprehensiveness, conciseness, robustness, and reliability, depending on their intended use Unterkalmsteiner and Abdeen (2023). However, our method abstracts existing, well-established taxonomies rather than creating new ones. The goal is to reduce granularity, which can enhance automatic classification performance. We conjecture that by merging overly specific categories into broader groupings with our proposed method, the resulting taxonomy becomes more practical for downstream tasks such as item classification. Therefore, our evaluation focus has been on classification performance. In our case study, since our classification setup was multi-label and included imbalanced categories, we reported both Micro-F1 and Macro-F1 scores, which are standard metrics capturing complementary aspects of performance in such problems. Micro-F1 reflects overall performance and gives more weight to classes with more examples, whereas Macro-F1 averages performance across labels, highlighting minority classes. Although we did not re-evaluate traditional quality metrics post-abstraction, expert oversight guided the refinement process.

The rest of this paper is organized as follows. Section 2 provides a review of background and related work on taxonomy generation and refinement aimed at reducing granularity and dimensionality. Section 3 is divided into two parts. The first part describes the proposed method for taxonomy generation, and the second outlines the evaluation process for the generated taxonomy, including details of the classification implementation. Section 4 validates the proposed method through a case study focused on creating a taxonomy of *innovations*. Section 5 presents the study's findings, while Sect. 6 discusses the results, the implications, and the limitations. Finally, Sect. 7 concludes the paper.

2 Background and Related Work

In this work, we proposed a semi-automatic method for creating taxonomies by leveraging the contextual knowledge and generative power of LLMs. We validated our method by using it to create an innovation taxonomy, which was previously developed through a manual process. Therefore, we have organized the related work into six main parts: (i) manual taxonomy creation methods, (ii) taxonomy refinement for reducing granularity and dimensionality, (iii) machine learning-based semi-automatic methods, (iv) symbolic and logic-based ontology learning, (v) knowledge graphs in ontology learning, and (vi) large language models (LLMs) in conceptual modeling.

2.1 Manual Taxonomy Creation Methods

Researchers have employed various methods for taxonomy creation, with the method proposed by Nickerson et al. (2013) being widely recognized in the literature for its systematic, transparent, and replicable framework. Kundisch et al. (2022) conducted a thorough study of research works that used Nickerson et al. (2013) method and proposed an update to the methodology. Their study revealed that about two-thirds of the taxonomies they analyzed followed Nickerson's methods. However, they noted that researchers continue to face challenges in reporting taxonomy development transparently and evaluating taxonomies effectively. To address these issues, Kundisch et al. (2022) introduced an updated methodology that includes 26 operational taxonomy design recommendations. Both the taxonomy creation methods by Nickerson et al. (2013) and Kundisch et al. (2022) follow an iterative process for building *flat* taxonomies from scratch. While their focus is on flat structures, many taxonomies in practice are *hierarchical*.

Some studies have adapted the Nickerson et al. (2013) method to develop hierarchical taxonomies. For example, Weking and Hein (2020) created a taxonomy of business model patterns using an inductive approach, bypassing the step of defining and refining the taxonomy based on a meta-characteristic, which is an important step of the Nickerson et al. (2013) method. They introduced stopping conditions based on specialization and generalization to reflect hierarchical relationships and applied agglomerative clustering to form an initial taxonomy. In a second iteration, they refined the clusters through expert review and split or merged them to establish hierarchical levels. While likely effective for organizing hierarchical entities as an initial step, their method remains heavily dependent on expert involvement, making it time-consuming.

We propose an adapted version of the method by Kundisch et al. (2022) for creating a *hierarchical* taxonomy. Our approach refines existing taxonomies by generalizing and consolidating categories to create a more abstract taxonomy. By incorporating LLMs, the method provides a more efficient alternative to manual taxonomy refinement, particularly in scenarios where time or expert resources are limited. We emphasize that the tasks performed by LLMs can also be carried out by experts, highlighting the flexibility of the proposed method.

2.2 Taxonomy Refinement for Reducing Granularity and Dimensionality

Recent studies have addressed the issue of high granularity (many levels) and dimensionality (many sibling groups) in hierarchical taxonomies by restructuring existing hierarchies. Our method aligns with this goal by generating more abstract representations through sibling merging or consolidation into parent groups (Wilsens et al. 2024). Wilsens et al. (2024) proposed a top-down clustering approach using entity embeddings to merge semantically close groups. According to the authors, their method is the first to support both merging of groups within the same hierarchical level and consolidation into parent nodes. While their method relies on embedding distances, our approach uses LLMs to assess semantic relatedness, drawing on contextual knowledge from large-scale corpora.

Other studies have explored data-driven signals for taxonomy adaptation. Babbar et al. (2016) pruned nodes using approximation error and generalization bounds, which indirectly reflect group size but require training a classifier at each node. We adopt a lighter approach by directly using group size as a proxy and relying on LLMs to assess semantic similarity among candidate groups. Their study showed that in domains with power-law distributed categories, selective pruning can improve classification performance compared to building taxonomies from scratch. In our method, we target long-tail categories and refine the hierarchy to support better classification, without retraining models.

For validation, we abstracted the CPC schema, extending prior work on patent taxonomy reduction. Babbar et al. (2016) applied SVMs with pruning heuristics to the IPC (~ 86K codes), an approach that would be computationally expensive for the CPC, which contains about 250K codes. Other methods, such as PatentSBERTa (Bekamiri et al. 2024), require fine-tuning large models and substantial GPU resources. Wilsens et al. (2024) proposed entity embeddings and top-down clustering, which involve additional threshold tuning and embedding calculations. In contrast, our method uses a single lightweight signal, class size, to identify candidate groups for merging, and leverages LLMs for semantic merging and label generation, achieving comparable classification performance to a manually constructed 83-label taxonomy with substantially less expert and computational effort.

In our recent study (Motamedi et al. 2024a), we presented a manual method to produce a more abstract representation of knowledge fields. Unlike approaches that truncate the hierarchy, this method considers document count as a proxy. The resulting taxonomy has three levels, with nine top-level sections and 83 lowest-level groups. In our case study, we used the same CPC schema to generate a taxonomy of innovations and compared it to the manually created one. The comparison focuses on how effectively each taxonomy meets the users' goals, which in this case study is classifying patents.

2.3 Machine Learning-Based Semi-Automatic Methods

Semi-automatic machine learning methods have been extensively used for ontology learning, although the most well-known tools cover the entire process of creating the ontology from data, they can also be used to adapt existing taxonomies. Text2Onto (Cimiano and Völker 2005) combined probabilistic logic with natural language processing (NLP) concept extraction to identify ontology elements from text. OntoLearn (Navigli et al. 2004) employed lexical resources alongside statistical learning techniques, later incorporating graph-based methods to support semantic structure induction (Velardi et al. 2013). OntoGen (Fortuna et al. 2006), primarily designed for constructing classification taxonomies, combined multiple text-mining algorithms within a graphical user interface. It supported bottom-up ontology development from data using SVM-based keyword extraction and k-means clustering, with both intra- and inter-cluster cosine similarity computed automatically.

However, these systems exhibit two primary limitations. First, they require continuous human supervision at each step (e.g., adding a concept) of the modeling process, limiting their degree of automation. Second, effective use of these tools often requires machine learning expertise, such as choosing appropriate values for n in n-grams or k in clustering, interpreting similarity metrics like cosine distance, and recognizing issues such as unrepresentative clusters or noisy keywords. This presents a significant barrier for domain experts who lack a background in machine learning.

The introduction of transformer-based models such as BERT (Devlin et al. 2019) improved specific aspects of taxonomy construction, particularly in reducing the need for supervision when organizing concepts hierarchically through *is-a* relationship inference (Chen et al. 2021). However, this approach still requires the generation of training data and fine-tuning of the BERT model beforehand.

In contrast to previous semi-automatic approaches, our method is designed to eliminate the barriers commonly associated with machine learning workflows. It does not require familiarity with algorithmic foundations, hyperparameter tuning, training data preparation, or fine-tuning of pre-trained models. This substantially lowers the entry threshold for domain experts, enabling them to engage in taxonomy refinement tasks without relying on machine learning engineers. The only user-defined inputs are the threshold factor, a scalar multiplier applied to the standard deviation, and the specification of conceptual elements such as the meta-characteristics, the target user, and the intended aim, which guide the perspective of the refined taxonomy. In our method, the expert may optionally review and refine the output at each iteration, providing adjustments as needed. That said, our method is not designed to completely build a taxonomy from scratch based on just data, but rather to adapt existing taxonomies. This fact reduces the need for techniques such as text clustering or relation mining from text.

2.4 Symbolic and Logic-Based Ontology Learning

Early research introduced the term *ontology learning* and proposed semi-automatic frameworks that combined statistical pattern recognition with Description Logic (DL) formalisms (Maedche and Staab 2001). DL-Learner (Lehmann 2009) integrated Inductive Logic Programming (ILP) with DL reasoning by adapting ILP techniques to operate within the constraints of DL for learning class expressions in the Web Ontology Language (OWL). AIMIE (Galárraga et al. 2013) and its extension AIMIE+ (Galárraga et al. 2015) offered effective approaches for mining logical rules from Knowledge Bases (KBs) and Knowledge Graphs (KGs) using ILP. Formal Concept Analysis (FCA) is another symbolic approach employed in ontology and taxonomy learning, functioning as a complementary method to DL. It was central to OntoComP (Sertkaya 2009), an interactive tool that leveraged FCA to engage ontology engineers and ensure the completeness of evolving OWL ontologies.

These ontology induction approaches prioritize formal correctness, relying on automatically mined rules that are supervised and refined by ontology engineers familiar with logic formalisms and the specific logic languages chosen. However, this dependence on ontology experts presents a major barrier to adoption, requiring significant up-front effort in rule definition, data structuring, or access to a well-formed KG. Over time, these methods have also incorporated statistical inference, creating hybrid reasoning approaches, NLP methods for extracting concepts from text data as in (Cimiano and Völker 2005), and machine learning techniques, further raising the expertise required for effective use. In contrast, LLMs offer a flexible, data-driven alternative that can interpret unstructured or loosely structured group descriptions without relying on formal rules or pre-existing KGs. Our method leverages LLMs' contextual understanding to assess semantic relatedness and propose group mergers, reducing the need for dedicated ontology engineers, logic formalization, manual rule pruning, or automatic methods for rule mining.

2.5 Knowledge Graphs in Ontology Learning

Another related line of work involves the use of KGs or KBs combined with hybrid reasoning to refine and extend ontologies and taxonomies (Li et al. 2020). Recent methods often integrate RDF-structured data and/or OWL rules with KG embeddings to support tasks such as completion and alignment (Chen et al. 2025). For example, Martel and Zouaq (2021) applied unsupervised hierarchical clustering to KG embeddings such as RDF2Vec (Ristoski and Paulheim 2016) to induce taxonomic hierarchies. In contrast, Pietrasik and Reformat (2020) proposed a method for inducing class taxonomies directly from KG triples using co-occurrence and frequency-based similarity, avoiding the need for embeddings. More recently, research has begun to combine description logic with LLMs, which provide background semantic knowledge for generating the embeddings themselves (Alam et al. 2025).

These KG-based and hybrid reasoning methods offer the advantage of leveraging structured, semantic representations to ensure logical consistency and facilitate large-scale taxonomy induction. However, they require suitable KGs, expertise in descriptive logic, symbolic reasoning, rule mining, embedding training, or statistical inference. Our LLM-centric approach addresses these limitations by enabling semantic consolidation without the need for structured data, pre-defined rules, embeddings, or formal ontology engineering skills. This aligns with ongoing efforts toward lightweight, expert-accessible refinement, while remaining compatible with future integration into KG-based verification frameworks for enhanced robustness.

2.6 Large Language Models in Conceptual Modeling

Recent research has explored the potential of Large Language Models (LLMs) in ontology learning, as summarized in the survey by Perera and Liu (2024). LLMs have demonstrated promise in automatically extracting and structuring knowledge for various ontology learning tasks across diverse domains (Giglou et al. 2023). For example, Silva et al. (2024) used structured prompts with LLMs (e.g., GPT, Claude) to generate ontologies, partially validating them through RDF syntax checks, OWL reasoning, and SHACL constraints, while experts curated task descriptions and prompt templates. Similarly, Fang et al. (2024) constructed a drug indication taxonomy from free-text labels using GPT-4 and real-world evidence (RWE), with LLMs extracting terms and inferring subsumption relations, and experts defining top-level categories and guiding prompt design. Their method, designed for building taxonomies from scratch through an empirical-to-conceptual process, remains computationally intensive and highly dependent on domain-specific data. In contrast, our approach abstracts an existing taxonomy using a lightweight signal, class size, to identify candidate groups for merging. These are then semantically evaluated by an LLM, reducing reliance on specific datasets and enabling broader applicability.

Beyond ontology construction, Kommineni et al. (2024) proposed semi-automatically populating a KG and utilizing an LLM to evaluate the quality of the generated KG as a form of automatic validation. In another study, Shah et al. (2025) presented a methodology for creating user intent taxonomies for web search analysis by employing LLMs to generate concepts and descriptions, with human experts validating the results. LLMs were used to generate candidate taxonomies from search and chat logs, including detailed descriptions and examples of intent categories, with minimal manual effort. Experts validated and refined the candidates and reviewed LLM-generated labels. Their study demonstrated that combining LLMs with expert input can efficiently produce high-quality taxonomies.

While our work also employs an LLM for taxonomy creation, it focuses on refining and modifying an existing taxonomy rather than generating one from scratch, and therefore assigns the LLM a different set of tasks compared to the previously mentioned studies. To this aim, and similar to prior work, we incorporated human experts to curate the task description as prompt templates and to review the outputs generated by the LLM.

Our method builds on the well-established taxonomy design process by Kundisch et al. (2022), adapted for hierarchical refinement of existing taxonomies using LLMs. Unlike prior semi-automatic approaches that rely on different data-driven signals and costly model training, our method uses class size as a lightweight signal and leverages LLMs for group merging and label generation, without requiring embeddings, classifiers, or external resources such as lexicons or KGs. Domain experts define key conceptual elements (e.g., purpose, target users, meta-characteristics), which are incorporated into prompts to guide LLM decisions. The method offers several benefits. First, it reduces required expertise by eliminating the need for training data, fine-tuning, or technical familiarity with machine learning or ontology engineering. Second, it retains flexibility through expert validation while automating core tasks such as semantic grouping and labeling. Third, it provides a transparent, replicable process aligned with established taxonomy frameworks and supports reproducibility across domains.

3 Materials and Methods

In line with hierarchical approaches, Chen et al. (2020) defines a taxonomy as a tree-based schema, comprising a set of concepts with specific relations and shared resources. In this work, we adopt Chen's hierarchical taxonomy definition.

Taxonomy. A taxonomy T_k = (C, R) is a tree structure where:

- C is the set of concepts in the taxonomy. Each concept c_i is named by its label, denoted as *label* (c_i), which is a string that may be composed of several words. Since the taxonomy is represented as a hierarchical representation, the topmost concept is referred to as c_{root}.

- R represents the relation between concept pairs. Different types of relations can be defined. In this work, we only consider *is_sup* (i.e., superior categories) and *is_sub* (i.e., subordinary categories). For a concept pair $c_i \leq c_j$, c_i has the relation *is_sub* to c_j, and c_j has the relation *is_sup* to c_i. The children in the tree structure are all having *is_sub* relation with their parents.

We adapted the taxonomy creation method proposed by Kundisch et al. (2022) to develop a hierarchical taxonomy using an existing hierarchical schema. Our proposed taxonomy generation method consists of four main steps: (i) identifying the problem and its motivations, (ii) defining the objectives of the solution, (iii) designing and developing, and (iv) evaluating the taxonomy. Each step of the process is discussed in detail in this section. Fig. 1 illustrates the proposed taxonomy generation method.

3.1 Identify the Problem and Motivate

Kundisch et al. (2022) outlines a three-step process for identifying the problem and its motivation, all of which are retained and remain relevant in our proposed method. These steps include identifying the phenomenon, defining the target user group(s), and understanding the users' purposes for using the taxonomy. By following these steps, the taxonomy creators should justify why a particular taxonomy serves as an appropriate conceptualization to meet the particular needs of the intended users. It is important to justify how the taxonomy will benefit the intended users in achieving their goals.

The purpose of the taxonomy for target users can be purely structural, aimed at organizing known constructs and their relationships related to the phenomenon under consideration. When generating a taxonomy from the existing structural schema, the purpose typically aligns with the structural goals of the original taxonomy while being tailored to fit the particular aims of the target users. All conceptual decisions, such as defining the phenomenon, identifying the target users, and specifying the task goals, should be made in collaboration with domain experts. Once these key elements are defined, they are embedded into the prompts used to guide the LLM. The LLM then uses this information to identify misaligned groups, suggest group merges, and generate representative labels for merged categories, as further detailed in subsequent sections.

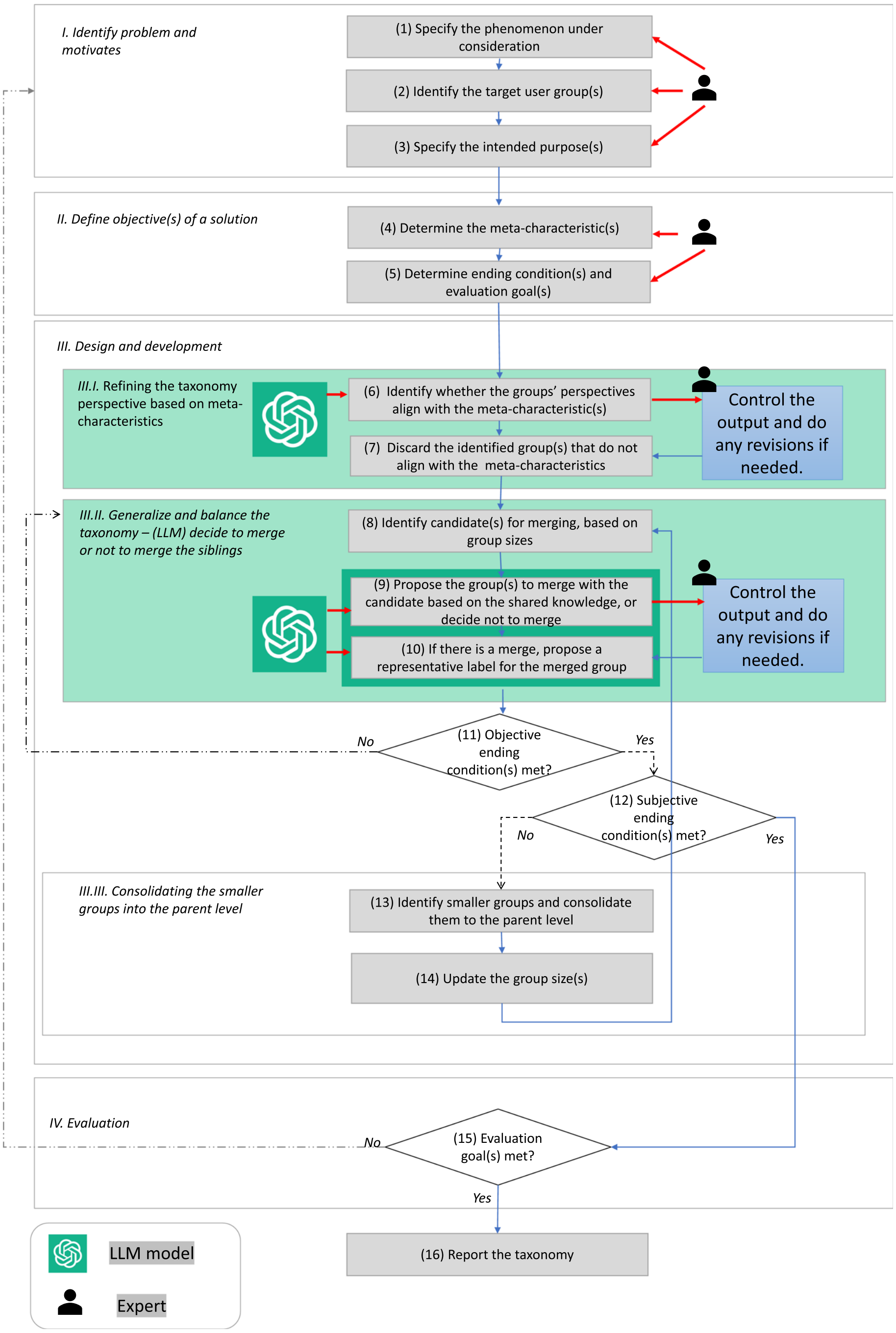
3.2 Define the Objectives of the Solution

After identifying the users and their purposes for using the taxonomy, the meta-characteristics should be determined. Meta-characteristics play an important role in the taxonomy development process as they define the perspective from which the phenomenon under consideration is analyzed to generate the taxonomy. When creating a taxonomy by leveraging existing structural schema, it is important to ensure that the constructs are relevant to the taxonomy design, a consideration guided by the definition of meta-characteristics. For instance, if we consider the creation of an *innovation taxonomy* for the target users, the resulting taxonomy can vary depending on the chosen meta-characteristics. If the meta-characteristic is determined as *knowledge field*, the taxonomy might include constructs such as *human necessities*, representing the innovations that are associated with essential needs such as food and clothing. On the other hand, if the meta-characteristic is determined as *innovative contribution*, the constructs might include categories such as *sustainable food production technologies* or *wearable health-monitoring devices*, reflecting particular advancements and the contributions of innovations. These variations highlight how the taxonomy structure shifts based on the perspective defined by the meta-characteristic.

The next step in the taxonomy generation method proposed by Kundisch et al. (2022) is defining the stopping conditions and evaluation goals. Since our method follows an iterative approach, stopping conditions should be defined to ensure the design process ends effectively. Additionally, defining evaluation goals early in the process is important, as these serve to guide the design and verify its success in the final stages. Evaluation goals should also be determined at this stage, which can be formed depending on the intended purpose of the taxonomy. For example, if the taxonomy is intended for classifying objects, the evaluation goal might focus on achieving accurate classification outcomes. On the other hand, if the purpose is to cluster objects based on similarities and differences, the evaluation goal could emphasize the degree of intra-group similarity compared to inter-group differences within the taxonomy. Defining meta-characteristics, ending conditions, and evaluation goals should be done in collaboration with domain experts.

Similar to Nickerson et al. (2013), our proposed method incorporates both objective and subjective ending conditions. As an iterative approach, the objective ending condition aims to terminate the process once particular criteria are met. For instance, the absence of any merge operations in the final iteration can serve as an objective ending condition. On the other hand, the process should assess whether the output satisfies users' expectations upon meeting the objective ending condition. This assessment is captured by the subjective ending condition, which evaluates subjective factors such as the level of abstraction achieved.

Fig. 1 Workflow of the proposed taxonomy abstraction method


I. Identify problem and motivates
(1) Specify the phenomenon under consideration
(2) Identify the target user group(s)
(3) Specify the intended purpose(s)
II. Define objective(s) of a solution
(4) Determine the meta-characteristic(s)
(5) Determine ending condition(s) and evaluation goal(s)
III. Design and development
III.I. Refining the taxonomy perspective based on meta-characteristics
(6) Identify whether the groups' perspectives align with the meta-characteristic(s)
(7) Discard the identified group(s) that do not align with the meta-characteristics
Control the output and do any revisions if needed.
III.II. Generalize and balance the taxonomy – (LLM) decide to merge or not to merge the siblings
(8) Identify candidate(s) for merging, based on group sizes
(9) Propose the group(s) to merge with the candidate based on the shared knowledge, or decide not to merge
(10) If there is a merge, propose a representative label for the merged group
Control the output and do any revisions if needed.
No
(11) Objective ending condition(s) met?
Yes
(12) Subjective ending condition(s) met?
No
Yes
III.III. Consolidating the smaller groups into the parent level
(13) Identify smaller groups and consolidate them to the parent level
(14) Update the group size(s)
IV. Evaluation
No
(15) Evaluation goal(s) met?
Yes
(16) Report the taxonomy
LLM model
Expert

3.3 Design and Development

While the previous steps are common to both manual taxonomy creation and semi-automatic refinement, the design and development step represents the key difference in our approach. We adapted this phase to enable the refinement of existing taxonomies, and this is also where we introduce the use of LLMs to support tasks such as identifying unaligned groups with the meta-characteristics, proposing merges, and generating representative labels. In contrast, the design step in the Nickerson et al. (2013) method involves building taxonomies from scratch through two parallel approaches: empirical-to-conceptual, where dimensions are derived from observed objects and their shared characteristics, and conceptual-to-empirical, where experts define dimensions first and then examine the objects accordingly. Our method focuses on refining existing structures rather than generating categories from the ground up. The subsequent steps, including the evaluation and reporting of the taxonomy, are also shared with the Nickerson et al. (2013) method.

The design of our proposed taxonomy generation method encompasses three main steps: (i) refining the taxonomy perspective based on meta-characteristics, (ii) abstracting and balancing the taxonomy based on shared knowledge and group sizes (performed over one or more iterations), and (iii) further abstracting the taxonomy by keeping the smaller groups in the higher level. We will elaborate on each of these main steps in detail and discuss how LLMs can be leveraged to automate these processes. It is important to emphasize that these design steps can also be performed manually with expert assistance, providing a flexible approach to taxonomy creation.

3.3.1 Refining the Taxonomy Perspective Based on Meta-Characteristics

An important aspect highlighted by Nickerson et al. (2013) is the selection of meta-characteristics as a foundational step in taxonomy development, which can be refined during the iterative process. Meta-characteristics represent the perspective or angle from which the taxonomy is developed and should closely align with its intended purpose. For example, as illustrated by Nickerson et al. (2013), if we consider the creation of a taxonomy for classifying computer platforms, the taxonomy's design must be different based on its intended purpose. If the goal is to differentiate platforms by processing power, relevant meta-characteristics would include hardware and software attributes such as CPU performance, memory capacity, and operating system efficiency. However, if the aim is to classify platforms by user interaction, the meta-characteristics would emphasize features such as the platform's user interface.

Defining the meta-characteristics emphasized by researchers is an important step in creating a taxonomy (Lauf et al. 2023). However, this step is often overlooked, and even when undertaken, the details are frequently not disclosed to readers. According to Kundisch et al. (2022), only 53% of their reviewed studies provided clear definitions of their meta-characteristics. In our method, which derives a taxonomy from existing ones, it is important that the source taxonomy shares the same or similar meta-characteristics. If these meta-characteristics are not clearly defined or unavailable, researchers must refine the taxonomy perspective to align it with the meta-characteristics established in earlier steps. To refine the hierarchical schema, each group must be evaluated to determine whether it aligns with the defined meta-characteristics. Constructs that do not fit should be discarded. This evaluation can be conducted with the help of expert knowledge or LLMs. LLMs can be leveraged by providing prompts that guide the model to evaluate whether a given group aligns with the meta-characteristic. For example, the LLM can be prompted as follows: $<<$ *Your task is to refine a taxonomy of [Domain Name], focusing on [Meta-Characteristic]. Based on your evaluation, decide whether to: (i) Retain the category if it aligns with the meta-characteristic, or (ii) Remove the category if it does not align.* $>>$ (the full set of prompts and implementation code is available online. See Sect. 8. Appendix). To provide the necessary context, each prompt includes the category label, its parent label, and its immediate children. We conjectured that LLMs can perform this task effectively due to their contextual understanding, which enables them to assess whether a construct aligns with the specified meta-characteristics. In our proposed method, we have included an expert review of LLM decisions to ensure that the discarded groups are indeed appropriate for exclusion.

3.3.2 Generalizing and Balancing the Taxonomy

Our method creates a more abstract taxonomy by building on an existing one and using a dataset in which items are assigned to their hierarchical groups. For example, in the CPC schema, each patent is linked to a group via its hierarchical code. A dataset of patents and their corresponding CPC codes enables the iterative abstraction phase of our method, where group size approximates development level. Groups with few objects are assumed to be less developed and, therefore, become candidates for merging during the abstraction process.

In this phase, only groups with relatively few assigned items, referred to as "long-tail" groups, are considered for merging. Researchers can define these groups using an objective criterion. As described in Algo. 1, a group can be flagged as a candidate for merging if its item count is below a threshold determined by the total number of items among its siblings or other groups at the same level. While this threshold may vary by design, we recommend adopting an objective approach to maintain consistency and generalizability. To identify long-tail groups in the taxonomy, we define a threshold as follows:

Algorithm 1 Iterative taxonomy abstraction through Long-tail group identification (Step III.II in Fig. 1). Decision steps, including evaluating shared knowledge between groups and assigning new labels, are automated via LLM prompting (GPT-4 in our case study).

```
**Input:** ;
#Selected with expert input;
Original taxonomy $T_{\mathrm{inp}}$ with hierarchical groups;
Dataset $D$ of items assigned to groups in $T$;
#Selected with expert input;
Threshold factor $n$ for identifying long-tail groups;
**Output:** Updated taxonomy $T_{\mathrm{out}}$
*Initialization:* $T = T_{\mathrm{inp}}$;
*Step 1: Compute threshold for all groups* ;
$S = \{$sibling groups, groups at the same level$\}$;
Compute the threshold $\tau = \max_{s \in S}(\mu(s) - n \cdot \sigma(s))$;
for each group $g \in T$ do
    if $size(g) < \tau$ then
        # identify Long-tail groups;
        Mark $g$ as a candidate for merging;
    end
end
*Step 2: Decide if the candidate should be merged with siblings*;
for each candidate group $c$ do
    if $c$ has siblings then
        # Via LLM prompting;
        Evaluate the shared knowledge between $c$ and its siblings;
        if $c$ is to be merged with sibling $s$ (based on shared knowledge) then
            Merge $c$ with $s$;
            # Via LLM prompting;
            Assign a representative label to the merged group;
        end
        else
            $c$ remains unchanged;
        end
    end
end
*Final Step:* $T_{\mathrm{out}} = T$;
#Expert revisions if needed;
The expert may review the merged groups and revise if necessary;
**Output:** $T$ (Updated taxonomy with merged groups)
```

$$\tau = \max_{s \in S}(\mu(s) - n(s) \cdot \sigma(s)) \tag{1}$$

where:

- τ: The threshold for identifying long-tail groups.
- $\mu(s)$: The average size of the candidate group and its related groups.
- $\sigma(s)$: The standard deviation of item counts for the candidate group and its related groups in the set.
- $n(s)$: A scalar factor that determines the degree of deviation in the set (e.g., $n = 1$ for one standard deviation).

The set S includes sibling groups and groups at the same hierarchical level. In our formula, the threshold is defined as the maximum of $\mu(s) - n \cdot \sigma(s)$ across these sets. By adopting the maximum value, we ensure that if a candidate group is identified as a long-tail group in any of the defined sets, it qualifies as a candidate for merging.

Table 1 illustrates this calculation. The *sibling-based threshold* uses only sibling groups, while the *level-based threshold* considers all groups at the same level. Corresponding flags indicate whether a group meets the long-tail condition under each threshold. The final flag is determined using the maximum of the two thresholds. For instance, G_2, G_3, and G_4 are not flagged for merging if the threshold is calculated only using sibling groups, but they are flagged for merging when using the level-based threshold. Conversely, H_6 is not flagged for merging if the level-based threshold is used, but is flagged for merging if sibling-based criteria are applied. The definition of the threshold may vary depending on the designer's goals and the characteristics of the taxonomy.

Our threshold formula provides an objective way to identify long-tail groups by subtracting $n \cdot \sigma(S)$ from the average size of related groups. This highlights groups that are relatively small and thus candidates for merging in the iterative abstraction process. The parameter n can be tuned based on the desired level of abstraction. For example, $n = 1$ captures moderately underdeveloped groups, while higher values ($n = 2$) target more extreme cases, reducing the number of merge candidates. This flexibility ensures the method can adapt to different taxonomies with different group sizes and the desired level of abstraction.

After identifying candidate groups for merging, each candidate is considered to determine whether it should be merged with a sibling group. LLM can assist in this process by the decision to merge, and if so, proposing a representative label for the merged group. Each iteration produces an intermediate output that can be reviewed by experts to ensure alignment with their expectations. Our method includes expert review of LLM merge decisions and revisions if needed. To assist with the merging decision, the prompt could be as follows: $<<$*Your task is to develop a taxonomy of* <Domain Name>, *focusing on <Meta-Characteristic> as the meta-characteristic. Decide to merge if the candidate category shares overlapping knowledge, has a similar meaning, or is closely related to one of the sibling categories. The goal is to reduce redundancy in the taxonomy; merge if combining similar categories leads to a more abstract and organized structure. Keep the candidate category separate if it represents a distinctly different field or concept from all siblings.* $>>$ Additionally, the LLM may be instructed on the expected output format. To provide the necessary context, the prompt includes the candidate category label, its parent label, and the labels of its sibling categories (the full set of prompts and implementation code are available online. See Sect. 8. Appendix).

If the shared knowledge between the candidate and any of its siblings is not substantial, the candidate group remains unchanged in this step. On the other hand, if a decision is on the merger of the candidate group with one of the siblings, a representative label will be assigned to the newly merged group. For example, to assist with generating representative labels, the prompt could be as follows: $<<$*Your task is to*

Table 1 Threshold comparison showing the impact of combining sibling- and level-based criteria

Parent group	Group	Item count	Sibling-based threshold	Sibling-based flag	Level-based threshold	Level-based flag	Final threshold	Final flag
G_H	G1	90	91.9	Yes	**126.1**	Yes	126.1	Yes
	G2	95	91.9	No	**126.1**	Yes	126.1	Yes
	G3	100	91.9	No	**126.1**	Yes	126.1	Yes
	G4	105	91.9	No	**126.1**	Yes	126.1	Yes
G_H	H1	1000	**690.6**	No	126.1	No	690.6	No
	H2	1100	**690.6**	No	126.1	No	690.6	No
	H3	1200	**690.6**	No	126.1	No	690.6	No
	H4	1300	**690.6**	No	126.1	No	690.6	No
	H5	1400	**690.6**	No	126.1	No	690.6	No
	H6	300	**690.6**	Yes	126.1	No	690.6	Yes

The table summarizes item counts, thresholds, and merge flags under sibling-based, level-based, and final (maximum) analyses. In this example, the parameter $n = 1$ is used in the threshold calculations *develop a taxonomy of <Domain Name>, focusing on <Meta-Characteristic> as meta-characteristic. You have to merge the following categories (i.e., the candidate group and sibling) to create a new, more abstract class in a taxonomy of knowledge fields. Suggest a representative label for this merged category. The label should be up to 10 words long and reflect the combined knowledge of the merged classes.* > > To provide the necessary context, the prompt includes the category label, sibling label, and the parent label of the category and the sibling (the full set of prompts and implementation code is available online [??]. Further details on the role of the LLM in these decisions will be provided in the following sections, where the case study will be discussed.

3.3.3 Evaluating Ending Conditions

After each iteration of abstracting and balancing the taxonomy, the ending conditions should be evaluated. First, the *objective ending condition* is assessed. If the objective ending condition is not satisfied, the abstracting and balancing step should be repeated. If the objective ending condition is satisfied (e.g., no merge occurred in the previous iteration), the *subjective ending condition* is then evaluated. The subjective ending condition reflects the goals and preferences of the users. For example, since the primary aim of this method is to abstract the taxonomy to make it more practical for the target users, the subjective ending condition can be defined as *the number of groups in the created taxonomy*, which should be defined through consultation with target users.

If the subjective ending condition is not met, the process continues to the next step of abstracting the taxonomy, where smaller groups are moved from the lower to the higher levels of the hierarchy. If the subjective ending condition is satisfied, the iterative process concludes and proceeds to the next step: assessing whether the evaluation goals have been met.

3.3.4 Consolidating the Smaller Groups into the Parent Level

Since the subjective ending condition may not have been met in the previous step, meaning that the level of abstraction is still insufficient (quantified as the number of groups in the taxonomy), it is necessary to further consolidate the groups into the parent level. This step aims to reduce the number of groups, ensuring the process effectively meets the subjective ending condition (see Algo. 2). To achieve this, some groups will be merged with their parents and placed one level higher in the hierarchy. Particularly:

- Leaf nodes with the smallest count among siblings.
- Non-leaf nodes with a single child and the smallest count among siblings.

After performing this step, the count and threshold values must be updated. Since this abstraction modifies the count values, the process transitions back to step III.II (see Fig. 1) to re-evaluate and identify any long-tailed groups that may emerge.

Algorithm 2 Taxonomy nodes consolidation by merging smaller groups into the parent level (Step III.III in).

```
Input: Taxonomy $T_{\mathrm{inp}}$;
Dataset $D$ of items assigned to groups in $T$;
Current level $CL$ (initialised as the leaf-node level);
Output: Updated taxonomy $T_{\mathrm{out}}$
Initialization: $T = T_{\mathrm{inp}}$, $CL =$ the leaf-node level;
for each node $c \in CL$ do
    if $c$ is a leaf node and count($c$) is the smallest among siblings then
        Merge $c$ with its parent node;
        Update the merged node's properties (e.g., count, threshold) in $T$;
    else if $c$ is a non-leaf node with a single child and count($c$) is the smallest among siblings
      then
        Merge $c$ with its parent node;
        Update the merged node's properties (e.g., count, threshold, children) in $T$;
    end
end
Update counts and thresholds for all modified nodes;
Recalculate group sizes and the threshold values;
Move to the next higher level: $CL =$ parent level(CL);
return $T_{\mathrm{out}} = T$ (Updated taxonomy)
```

3.4 Taxonomy Evaluation

When both objective and subjective ending conditions are met, the taxonomy generation process concludes. At this point, the predefined evaluation goals should be assessed to determine how well the taxonomy meets users' intended objectives. Kundisch et al. (2022) observed that only one-third of taxonomies in their review were evaluated, leading them to propose ten evaluation methods tailored to various research challenges, along with corresponding goals and guidelines. The chosen evaluation metric should align with the taxonomy's intended use. For classification tasks, metrics evaluating the efficiency and accuracy of classification may be most relevant. On the other hand, if the taxonomy is intended for clustering or exploring objects, measures such as intra- and inter-class similarity are appropriate.

4 Case Study: Innovations Taxonomy

Using our proposed method, we developed a taxonomy of *innovations* based on the detailed CPC taxonomy for patent documents. After describing the dataset, we outlined each step of the process, highlighting how our method ensures thorough documentation of all steps and their outputs. We also present evaluation metrics and compare the results to a manually created taxonomy of knowledge fields. Table 2 summarizes each step, with details specific to the *innovation* taxonomy.

4.1 Patents in CPC System

Innovations are disseminated through various channels, including academic publications, development platforms (e.g., GitHub), media, and patent documents. Patents, as defined by the World Intellectual Property Organization (WIPO), grant inventors exclusive rights to their creations.[1] This work focuses on patent data as a key source of innovation-related information.

To facilitate search and retrieval, systems such as the CPC, an extension of the International Patent Classification (IPC), assign unique hierarchical codes to patents (Gomez and Moens 2014; Lee et al. 2021). Introduced in 2013, CPC has seen growing adoption and now covers a wide range of patents globally (Lee and Hsiang 2020). The hierarchy consists of nine top-level sections, divided into classes, subclasses, groups, and subgroups. The CPC scheme is publicly available online[2]. An example of the CPC hierarchical structure is shown in Table 3.

The CPC hierarchy includes approximately 250,000 classification entities (Kamateri et al. 2024), with highly detailed labels at the lowest level. In our previous work, we manually generalized several groups within the taxonomy to create a more abstract version (Motamedi et al. 2024a). We compare the taxonomy generated using our proposed method with the manually developed taxonomy with 83 categories at the lowest level. Similar to our case study, the goal of manually creating a taxonomy was to assist users in classifying innovation-related documents by knowledge fields.

4.2 Data Preparation

The CPC dataset used in this study is the Google Patents Public Datasets, accessible via BigQuery [3]. Each record in the dataset corresponds to a patent application and includes details such as the publication number, application number, CPC code, patent title, abstract, and description. We have expanded the dataset to include the titles associated with each CPC code from Espacenet.[4] The abstract is a short text that highlights the innovative aspect of the patent, while the description provides a more comprehensive account, potentially including related work and additional technical details. To generate the input text for this study, we concatenated the title, followed by the abstract, and then the description. Only documents with a concatenated text length of at least 100 words were included, following prior research that demonstrates improved classification performance with the first 100 words (Li et al. 2018).

The taxonomy creation method includes counting the number of documents in each generated group. Prior to counting, preprocessing was conducted to ensure consistency and comparability with the manually created taxonomy. Preprocessing included deduplication to identify and remove both exact and near-duplicate textual entries (Costa et al. 2011; Lee et al. 2022). We used MinHash-based Locality Sensitive Hashing (LSH), a robust method for detecting similar documents (Gyawali et al. 2020; Jafari et al. 2021; Wang et al. 2020), with a similarity threshold set at 0.9. Hash signatures were generated using 128 hash permutations, and the method was applied to n-gram tokens ranging from 1-grams to 3-grams.

[1] https://www.wipo.int/portal/en/ (accessed 02 Dec 2025).

[2] https://www.cooperativepatentclassification.org (accessed 02 Dec 2025).

[3] https://github.com/google/patents-public-data (accessed 02 Dec 2025)

[4] https://worldwide.espacenet.com/ (accessed 02 Dec 2025)

Table 2 Taxonomy Design Recommendations (Rec) in the refinement of a taxonomy for *innovations*

Activities	Step(s)	Design Rec (actor)	In *innovation* taxonomy
Identify problem and motivate	1–3	Specify the phenomenon under consideration (expert)	Phenomenon: innovations
	1–3	Specify the taxonomy's purpose(s) (expert)	Purpose: To assist target users in tracking innovations and classifying innovation-related documents
	1–3	Specify the taxonomy's target user group(s) (expert)	Target users: researchers, practitioners, and policymakers
Define objectives of a solution	4–5	Determine the meta-characteristic(s) (expert)	Meta-characteristic: knowledge fields
	4–5	Determine the ending conditions and evaluation goal(s) (expert)	Objective ending condition: No new merge in the previous iteration; Subjective ending condition: The number of leaf-node groups <100; Evaluation goal: Classification of patents (Micro-F1 score and Macro-F1 score)
Design and development	6–14	Identify whether the groups' perspectives align with the meta-characteristic(s) (LLM)	
	6–14	Revise, if needed, the identified unaligned group(s) (expert)	The expert did not decide on any revisions
	6–14	Discard the identified unaligned group(s)	Discarded 41 unaligned groups
	6–14	Identify candidate(s) for merging, based on group sizes	n=1 in Eq. 1
	6–14	Propose the group(s) to merge with the candidate based on the shared knowledge, or decide not to merge (LLM)	
	6–14	Revise, if needed, the merged group(s) (expert)	The expert did not decide on any revisions
	6–14	If there is a merge, propose a representative label for the merged group (LLM)	
	6–14	Identify smaller groups and consolidate them into the parent level	
	6–14	Update the group size(s)	
IV) Evaluation	15	Check if the evaluation goal(s) is met	Evaluated for its effectiveness in classifying patents
Report the taxonomy	16	Report the taxonomy	

4.3 Innovations Taxonomy: Creation Process

Researchers, practitioners, and policymakers can benefit from a taxonomy of innovations that organizes the vast textual data produced across media, enabling them to track developments within and across fields. To build such a taxonomy, we leveraged the widely used CPC patent taxonomy, which categorizes applications by field of knowledge through a five-level hierarchy. The first level contains nine main sections that branch into increasingly detailed categories. However, the large number of categories makes the full schema impractical for supporting user tasks such as classifying items into categories. This motivated the need for a more abstract taxonomy that still represents core knowledge fields.

Identify the Problem and Motivate This case study focuses on *innovations* as the phenomenon under consideration. Our goal is to create a taxonomy of innovations that addresses the needs of the target users, *researchers, practitioners, and policymakers*, by helping them classify relevant documents across platforms such as patents, scientific articles, technological reports, and news.

Table 3 Example of the CPC code structure, illustrating levels from section to subgroup

Level	Code	Title
Section	A	Human necessities
Class	A42	Headwear
Subclass	A42B	Hats; head coverings
Group	A42B1/00	Hats; caps; hoods
Subgroup	A42B1/203	Inflatable

This organization follows a hierarchical structure and it is intended to *assist target users in tracking innovations and classifying innovation-related documents*.

Define the Objectives of the Solution In consultation with target users, we carefully selected the meta-characteristic, as it could substantially impact the taxonomy's usefulness and alignment with user expectations. We identified two key dimensions: (1) the application field where the innovation is applied, and (2) the scientific field contributing to its novelty. Based on these, we defined *knowledge fields* as the meta-characteristic for this case study. For the objective ending condition, we defined *no new merge in the previous iteration*, and for the subjective ending condition, we defined it as *the total number of leaf-node groups being fewer than 100*. The threshold was determined in consultation with target users to create a more practical taxonomy.

The evaluation goal of this study was *to have a classifier capable of classifying patent documents in text format into the groups of the generated taxonomy.*. We reported the *Micro-F1* and *Macro-F1* scores. Each patent document is associated with one or more groups in the taxonomy, making the task a multi-label classification task. The F1-score is a common metric for classification tasks, particularly in multi-label classification. We reported both Micro-F1, averaged across all instances, and Macro-F1, averaged across all classes.

Design and Development – Refining the Taxonomy To evaluate whether each group in the existing taxonomy aligns with the defined meta-characteristic (i.e., *knowledge fields*), we leveraged the GPT-4 model of LLMs. The model was prompted (the full set of prompts and implementation code is available online. See Sect. 8. Appendix) to identify groups that did not align with the meta-characteristic. Any groups flagged by the model as misaligned were subsequently reviewed and removed from the taxonomy. To ensure the validity of these decisions, an expert conducted a secondary review to confirm the model's assessments. The refinement process removed 41 groups from the hierarchy, resulting in a refined taxonomy with 606 groups at the lowest level (i.e., third level).

To illustrate the impact of considering "knowledge fields" as meta-characteristics, we provide examples of groups that were removed or retained based on their alignment with the defined meta-characteristic. The model identified groups such as *"artificial flowers, wigs, masks, feathers"*, *"hats, head coverings" and "small arms, e.g., pistols, rifles"* for removal. This decision was confirmed by an expert, who validated that these groups are product-oriented and lack the abstraction required to represent comprehensive knowledge fields. In contrast, groups such as *"planting, sowing, fertilizing"*, "furniture specially adapted for children", and *"separating solid materials using liquids or pneumatic tables or jigs"* were retained. According to the expert, these groups represent broader domains of applied knowledge, including agricultural practices, ergonomic design, and industrial engineering, each reflecting innovation and expertise within a defined knowledge field. A complete list of removed and retained groups along with their labels is available online (see Sect. 8. Appendix).

Design and Development – Generalizing and Balancing the Taxonomy In this step, it was necessary to count the number of documents assigned to each group from the refined taxonomy, across all levels of the hierarchy. As explained earlier, we used the Google Patents Public Datasets available on BigQuery [5] and performed the preprocessing outlined in Sect. 4.2 before proceeding to the next step.

We calculated the number of preprocessed documents in each category and enriched the taxonomy nodes with this information, along with a threshold derived from Equation 1, using parameter $n = 1$. Specifically, Equation 1 provides a threshold used by Algo. 1 to select candidate nodes for merging. Algo. 1 is an iterative algorithm designed to refine the taxonomy. In each iteration, it identifies candidates (i.e., nodes with a document count below the threshold) and sends their information to GPT-4 via a prompt. The model receives the labels of the candidate's siblings and parents and determines whether merging is appropriate (Step 2 of Algo. 1). If merging is recommended, a second prompt is used to generate a representative label for the merged group. The groups are then merged: their parent pointer is updated, children are combined, and counts are summed (the full set of prompts and implementation code is available online. See Sect. 8. Appendix)

Design and Development – Evaluating Ending Conditions We defined the objective ending condition as the point at which no new merges occur in the previous iteration. The subjective ending condition was defined as the point at which the taxonomy has fewer than 100 leaf nodes. If the subjective ending condition is not met, the process proceeds to the next step, where another round of abstraction is performed.

Design and Development – Consolidating the Smaller Groups Into the Parent Level In this step, we applied an abstraction algorithm, detailed in Algo. 2. The algorithm begins by identifying leaf nodes with the lowest document count among their siblings. The LLM model (GPT-4) is then prompted to determine which sibling has the most overlapping knowledge or shares a closely related

[5] https://github.com/google/patents-public-data (accessed 02 Dec 2025).

knowledge area. The LLM is instructed to select one sibling for merging and is not allowed to opt out of the merge.

If a leaf node has no siblings, the algorithm checks its parent node at a higher level. If the parent has the lowest count among its siblings, it is flagged for merging. If the parent node also has no siblings, the algorithm moves up one level to its parent and repeats the process. Once a sibling is selected for merging, the candidate node, its selected sibling, and their parent are sent in a subsequent prompt to the LLM model (GPT-4) to request a representative label for the merged group. The groups are then merged, updating the parent pointer, the children of the merged groups, and the document count.

Algo. 2 is executed for one round. Since the balance of the groups, which was addressed in the previous iterative algorithm (Algo. 1), may shift (as discussed in Sect. 3), the process returns to Step III.II in .

Taxonomy Evaluation Ultimately, when the subjective termination condition is met, defined as the taxonomy containing fewer than 100 groups, it is evaluated for its effectiveness in classifying patent documents. Additional details on the problem's classification formulation are provided in the following section.

4.4 Classification Methodology

We sampled the preprocessed data (see Sect. 4.2) for use in classification. Since this is a multi-label classification problem, achieving a completely balanced dataset is not feasible. However, to ensure a minimum representation of the smaller classes, we performed random sampling while considering the number of samples in each class. We set a minimum threshold of 2000 samples. Once a class reached this threshold, we stopped collecting additional data for that class to prioritize sampling from smaller classes. However, given the multi-label nature of the problem, selecting a smaller class for inclusion in the sample may also lead to the inclusion of a larger class if both labels are assigned to the same text. Thus, while we mitigated the imbalance, we did not fully address it because of the nature of the problem.

We divided the sampled data into training, validation, and test sets using a ratio of 80%, 10%, and 10%, respectively. To ensure consistent class distribution across these subsets, we used a multi-label stratified splitting approach,[6] to maintain the proportional representation of each class in all three sets. The same sampling and stratifying technique was applied to both the data used for our taxonomy and the manually created taxonomy. The number of items in the training, validation, and test sets is provided in Table 4.

In the patent dataset, each patent can be assigned to multiple CPC codes since the innovation can be related to several knowledge fields at the same time. Therefore, we approached the classification task as a multi-label problem, allowing each document to be associated with multiple knowledge fields. We aimed to classify patents into categories at the lowest level of our reported taxonomy. Considering the large size of the dataset, we decided to leverage pre-trained language models. Particularly, we used *distilroberta-base*, a lightweight version of RoBERTa (Liu et al. 2019; Sanh et al. 2019).

To tailor the pre-trained models for our classification task, we fine-tuned them by adding a classification head that uses the hidden state of the model's first token as input. The input is processed through a fully connected linear layer. To improve regularization and prevent overfitting, a dropout mechanism is applied, followed by a tanh activation function to introduce non-linearity. Given the multi-label nature of the task, the model's output logits for each class were converted into probabilities using a sigmoid activation function. For model training, we used a learning rate of $4e-5$ with a linear scheduler and a weight decay of 0.1. To mitigate overfitting, the best checkpoint was chosen based on evaluation metrics on the validation set. The model was trained for 10 epochs, with early stopping based on F1-score. The implementation of the classification task is available online (See Sect. 8. Appendix).

5 Results and Analysis

In this section, the results are presented in two parts. First, we introduce the taxonomy generated using our proposed method. Next, we compare the performance of classifiers in categorizing patents into the fine-grained classes of this taxonomy with that of the manually created taxonomy. The task was to classify patent documents into groups at the lowest level of the hierarchical structure.

5.1 The Taxonomy of Innovations

Using the proposed method, we generated a taxonomy of innovations with three hierarchical levels, without considering the root node. Table 5 compares some of the structural features of the taxonomy created using our method with the manually generated taxonomy, as well as the initial taxonomy used in our method after filtering out irrelevant groups based on meta-characteristics. As shown, the proposed taxonomy contains 8 primary classes at the first level, which are further divided into 48 groups at the second level and 36 groups at the third level, resulting in a

[6] https://github.com/trent-b/iterative-stratification?tab=readme-ov-file#multilabelstratifiedkfold (accessed 02 Dec 2025)

Table 4 Number of items in the train, validation, and test sets for the datasets used for classification

Method	Train set	Validation set	Test set	Sample dataset
Manual method	995,451	124,470	124,548	1,244,469
Automatic method	1,048,119	131,009	131,151	1,310,279

Table 5 High-level comparison of the taxonomies. The root node is not counted in the number of levels. Standard deviations are indicated in parentheses

Metric	Taxonomy (auto)	Taxonomy(manual)	Taxonomy(initial)
Number of levels	3	4	3
Number of leaf-node groups	93	83	606
Number of groups at level 1 to 4	8-48-36-0	9-28-54-14	8-118-599-0
Average number of children per parent node	6.6	6	6
Average Path to Leaf Nodes (std)	2.3 (0.6)	3 (0.6)	3(0)

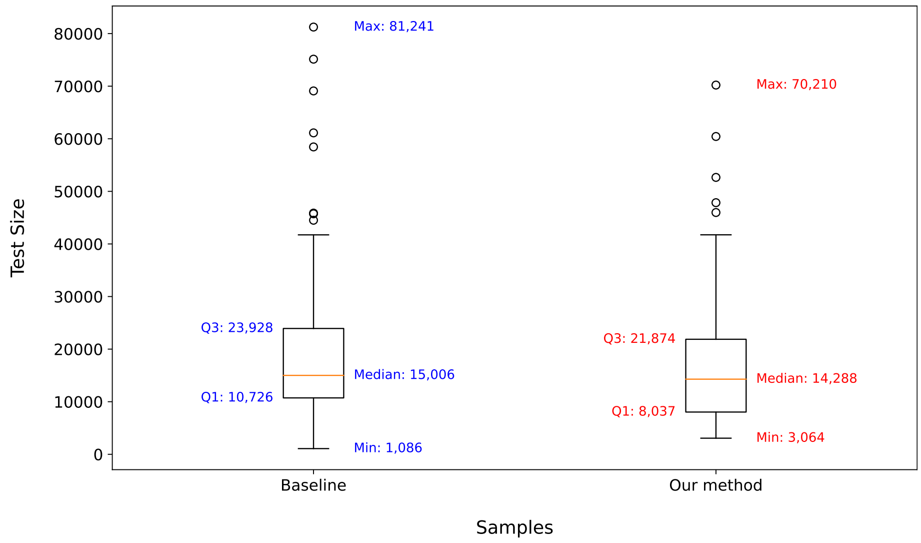


Fig. 2 Distribution of sample counts per class in test sets generated using our method compared to the baseline

total of 93 leaf-node groups (i.e., groups with no children). The manually created taxonomy includes 4 levels, with 83 leaf-node groups.

Our taxonomy demonstrates a slightly higher average number of children per parent node (i.e., 6.6 compared to 6). The average path length to leaf nodes is slightly shorter in our taxonomy (i.e., 2.3 levels with a standard deviation of 0.6) compared to the manually created taxonomy (3 levels with a standard deviation of 0.6). Overall, based on an expert review of the created taxonomies and the structural metrics presented in Table 5, both taxonomies exhibit similar structural properties, overlap in the created groups, and provide comparable coverage of various innovation fields. The created taxonomy, along with all intermediate outputs generated at each iteration of the method, is available online (See Sect. 8. Appendix).

Table 6 Classification results for two taxonomies, presenting performance metrics

Taxonomy	Macro-F1 score		Micro-F1 score	
	Val	Test	Val	Test
Taxonomy (auto)	0.93	**0.87**	0.90	0.70
Taxonomy (manual)	0.93	0.86	0.91	**0.71**

As part of the process, counting the number of documents in each group during the design and development phase is a necessary step. The balancing and abstraction steps in our method aim to create a taxonomy where groups have a comparable number of documents assigned, approximating a similar level of abstraction across groups.

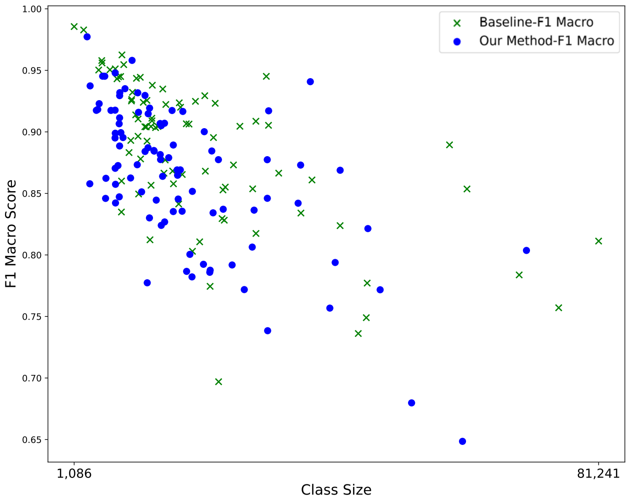


Fig. 3 Distribution of sample counts per class for test sets generated by the two methods: (i) the baseline and (ii) our method

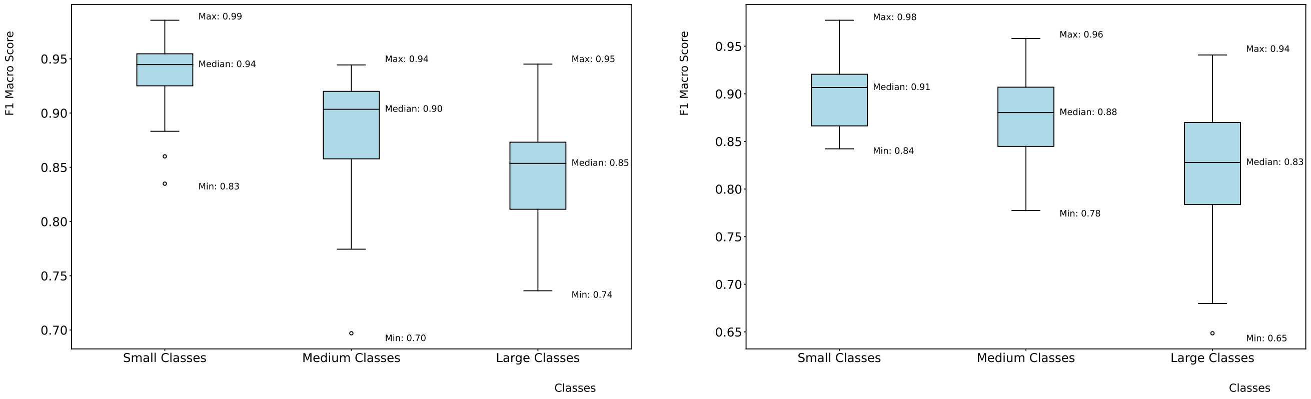
Fig. 3 presents a box plot comparison of class sizes in the baseline taxonomy (i.e., manually created) and the taxonomy generated using our method. Both taxonomies exhibit a reasonable variation in scale when comparing the minimum and maximum number of instances per group (Fig. 2).

5.2 Classification Results

When generating the taxonomy, the evaluation goal was to use the taxonomy for classifying objects (i.e., patents in this study). To understand whether the taxonomy was able to meet this goal, we formulated the problem as a multi-label problem and trained a classification model (see Sect. 4.4 for more details). The classification results, measured using Micro-F1 and Macro-F1 scores, are presented in Table 6. We presented the results when training the model for 10 epochs. We observe that our approach achieved comparable results to the manually created taxonomy.

As shown in Table 6, both the taxonomies generated by our method and the manually created one exhibit higher Macro-F1 scores compared to Micro-F1 scores. This suggests that the model performs relatively better on minority classes than on majority classes. To further analyze these results, we generated a plot (see Fig. 3) illustrating the relationship between Macro-F1 scores and the corresponding class sizes. The plot reveals that Macro-F1 scores are generally higher for minority classes, highlighting the imbalanced nature of the dataset. While we have addressed the imbalance to some extent and previously noted that the multi-label nature of the problem limits our ability to achieve a completely balanced dataset, we acknowledge that alternative approaches for mitigating dataset imbalance in multi-label problems could be explored to further improve the results.

From Fig. 3, we observe that classifying documents associated with smaller groups achieved higher F1-macro scores compared to larger groups. To further analyze this observation, we categorized all classes into three size groups: (i) small classes (first quartile), (ii) medium classes (second and third quartiles), and (iii) large classes (above the third quartile). Using this categorization, we generated a box plot of F1-macro scores across different class size categories, as shown in Fig. 4. The plot shows that, for both taxonomies, the median F1-macro scores for small classes are consistently higher than those for medium classes, and the median F1-macro scores for medium classes are higher than those for large classes. Medium classes show slightly lower medians compared to small classes and exhibit increased variability, while large classes display the lowest medians and the widest range of F1-



(a) Dataset compiled from manually created method

(b) Dataset compiled from our proposed automatic method

Fig. 4 Box plots of F1-Macro scores for small, medium, and large classes

Table 7 Intra-similarity, inter-similarity, and their ratio for different label combinations. The values in parentheses represent the standard deviation

Taxonomy	Average Intra-similarity	Average Inter-similarity	Inter/Intra Ratio
Taxonomy (auto)	0.60 (0.16)	0.17 (0.05)	0.28
Taxonomy (manual)	0.60 (0.17)	0.18 (0.04)	0.30

macro scores. Notably, the dataset generated using our method shows a smaller disparity in F1-macro scores between small and medium classes. However, for larger classes, the minimum F1-macro score is lower compared to the manually created one.

To evaluate the effectiveness of the taxonomy in organizing input text into well-separated groups in the embedding space, we extracted embeddings from the model's last layer for each document in the test set. These embeddings were normalized to ensure consistency in similarity measurements. For each group, we computed the intra-similarity, which reflects the closeness of vectors within the group, and the inter-similarity, which measures the closeness of vectors between different groups. We assumed that higher intra-similarity within a group, combined with lower inter-similarity between groups, suggests that similar patents are mapped to proximate locations in the embedding space, reflecting well-defined group boundaries. The results of this analysis, comparing the taxonomy generated using our proposed method with the manually created one, are presented in Table 7. The comparison shows that both taxonomies produce comparable intra- and inter-similarity values. Specifically, an intra-similarity value of approximately 0.6 and inter-similarity values of 0.17 or 0.18 suggest that documents within the same group are mapped to relatively close positions in the embedding space compared to documents in other groups.

6 Discussion and Future Work

The comparable intra- (0.6) and inter-similarity (0.3) between the taxonomies generated by our proposed method on the one hand, and the manually created one on the other hand, showed that documents within the same group were mapped to proximate positions in the embedding space while maintaining a larger distance between other groups. These results validated the structural soundness of the generated taxonomy and its ability to effectively cluster related documents, similar to the manually created taxonomy. Moreover, the number of groups at each level was less than or equal to that in the original taxonomy, the average root-to-leaf path length, and the maximum depth were reduced. Together, these findings answered the first research question of this study.

While improving the classification pipeline could yield more generalizable results, the primary goal of this study was to compare the effectiveness of our proposed taxonomy generation method against a manually created one. Using the same pipeline and dataset for both, we showed that their performance is comparable under a consistent study design. The results addressed the second research question, demonstrating that the proposed method achieved classification performance comparable *to* the manually created taxonomy in the downstream patent classification task.

The taxonomy generated using our method proved to be comparable to the manually created one. However, it showed greater consistency across class sizes, especially for small and medium classes (see Fig. 3). For large classes, our method produced a wider range between minimum and maximum Macro-F1 scores, suggesting it helps mitigate imbalance in smaller groups but may require refinement for larger ones. This challenge underscores the need for additional sampling strategies to achieve a more balanced dataset. The higher performance in smaller classes suggests that label quality in larger classes may require closer evaluation.

Demonstrating that our method can leverage existing detailed taxonomies to generate more abstract ones has important implications. Given the widespread availability of such structured resources, our LLM-based approach substantially reduces the need for expert involvement throughout the taxonomy creation process. This makes it especially practical in resource-constrained settings. By automating several steps, the method enables more efficient and scalable taxonomy generation. It also shifts the expert's role from manually designing taxonomies to overseeing and refining an automated process.

The method's ability to refine existing schema based on clearly defined meta-characteristics, combined with the adaptation of previously established methods, ensures its applicability across various domains and user needs. In the presented case study, we used only the CPC taxonomy. However, we acknowledge that multiple taxonomies often coexist both within and across domains. For example, patents are also organized using the International Patent

Classification (IPC).[7] Additionally, taxonomies developed for different purposes may exhibit overlapping groupings. One such example is the Standard Industrial Classification (SIC),[8] used by economic and statistical agencies to classify firms by sector, which may conceptually overlap with CPC categories. Although our current demonstration is limited to CPC, the proposed method is generalizable to other structured taxonomies across domains. As future work, following an alignment step between overlapping taxonomies, our abstraction method can be applied to the aligned nodes to generate higher-level representations. This would support the training of multi-source classifiers that leverage data from both patent and industry perspectives. Applying our method on top of an alignment process could offer better insights into its applicability across diverse taxonomic systems.

Unlike many existing taxonomy creation methods, particularly manual approaches, our method emphasizes transparency and evaluability. Through a case study, we validated the proposed method as a comprehensive framework for creating a taxonomy, evaluating the resulting taxonomy, and thoroughly documenting the process. The case study highlighted how the method enables detailed reporting of each step, ensuring a transparent, reproducible, and evaluable process. We compared the taxonomy created by our proposed method to the manually created one based on performance in a downstream classification task. Other quality metrics, such as completeness and correctness, were not re-evaluated in this study. However, given the clarity of the method and the available implementation, it is expected to be easily applicable to other domain taxonomies, where more formal evaluations using such metrics could be incorporated as part of future work.

While the proposed method offers several advantages, it also has some limitations that require attention. The reliance on LLMs can introduce biases that were not fully explored in this study. LLM outputs can occasionally reflect semantic drift or inconsistent reasoning, particularly in edge cases or poorly defined categories. While expert oversight mitigates this, the quality of the results may depend on the clarity of the prompts and the domain alignment of the document similarity model. We did not systematically study the variance of LLM responses. Informal spot checks suggested that merge decisions were generally stable across runs, with human oversight providing an additional quality check. Nonetheless, a systematic analysis of variance (e.g., issuing multiple calls per merge decision and quantifying agreement) remains an important direction for future work.

Moreover, our evaluation in the case study was limited to CPC patent data. Results may differ in other domains or taxonomies with different granularity and labeling practices. Additionally, CPC labels may contain noise, potentially affecting both merging decisions and evaluation. Our evaluation focused on structural measures and classification metrics and did not assess expert usability, interpretability, or temporal stability, which may yield additional insights. Scalability is affected by model versions and the computational cost of LLM calls and embeddings, which may impact large-scale deployments. Future work could examine robustness across embedding models and extend to additional domains and taxonomies, and assess usability with domain experts to support more generalized conclusions.

We classified the patent documents into the classes at the lowest level. Future work could explore classification algorithms that consider the hierarchical structure of classes in their mechanism, which may improve the results.

7 Conclusions

In this paper, we proposed a semi-automatic method for creating hierarchical taxonomies by generalizing overly detailed categories in taxonomies to better align with users' needs. The process leveraged the contextual knowledge and generative capabilities of LLMs. Our method processed an existing, overly detailed taxonomy, generalizing and consolidating several of its groups into the parent level by prioritizing the groups with fewer associated items. By validating the method through its application in creating an *innovation* taxonomy as a case study, we demonstrated that the resulting taxonomy performed comparably to a manually created one in meeting evaluation objectives. In our proposed method, LLMs were instructed to determine whether groups should be merged, and if so, which groups to merge, while also suggesting representative labels for the consolidated groups through an iterative approach.

It is clear from our work that LLMs can act as semantic proxies in taxonomy abstraction, enabling expert-guided but LLM-assisted refinement that scales more efficiently than traditional manual, ML, or logic-based approaches. What differentiates our method is its ability to abstract and consolidate taxonomies without requiring training specific models, formal rules, or symbolic KGs.

The case study showed that the classifier's performance using the taxonomy created by our method was comparable to that of the manually created taxonomy. Furthermore, our method guarantees that critical steps in creating taxonomies are documented and reported, ensuring

[7] https://www.wipo.int/en/web/classification-ipc (accessed 02 Dec 2025).

[8] https://www.sec.gov/search-filings/standard-industrial-classification-sic-code-list (accessed 02 Dec 2025).

transparency and reproducibility. To support further research, we have made the generated taxonomies and all outputs from our taxonomy generation method available online.

This study demonstrated that the use of LLMs can substantially reduce the taxonomy creation workload, shifting the experts' focus from building taxonomies from scratch to managing and refining outputs within an automated method. Our proposed method enables a more efficient and practical way to develop taxonomies, reducing time and resource requirements while maintaining quality by making better use of experts throughout the process.

Appendix

All experiments were performed using Python 3.11.0 on Windows. The proposed taxonomy abstraction pipeline accessed OpenAI's GPT-4 model through the official API interface (Chat Completions endpoint). Default sampling parameters were used: temperature = 1.0, top_p = 1.0, presence_penalty = 0.0, and frequency_penalty = 0.0 (as of 2025-06-02). Prompt–response pairs were cached locally across runs.

Online Appendix: Code, prompt templates, input, and outputs are publicly available at: https://github.com/elmotamedi/TaxoRefine (accessed 02 Dec 2025).

Acknowledgements This research was funded through the projects DataPACT (HE 101189771) and enRichMyData (HE 101070284).

References

Alam M, van Harmelen F, Acosta M (2025) Towards semantically enriched embeddings for knowledge graph completion. Neurosymbol Artific Intell 1(NAI–240):731. https://doi.org/10.3233/NAI-240731

Babbar R, Partalas I, Gaussier E, Amini MR, Amblard C (2016) Learning taxonomy adaptation in large-scale classification. J Mach Learn Res 17:1–37

Bekamiri H, Hain DS, Jurowetzki R (2024) PatentSBERTa: a deep NLP based hybrid model for patent distance and classification using augmented SBERT. Technol Forecast Soc Chang 206(123):536. https://doi.org/10.1016/j.techfore.2024.123536

Chen C, Lin K, Klein D (2021) Constructing taxonomies from pretrained language models. In: Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies, Association for Computational Linguistics, Online, pp 4687–4700, https://doi.org/10.18653/v1/2021.naacl-main.373

Chen J, Mashkova O, Zhapa-Camacho F, Hoehndorf R, He Y, Horrocks I (2025) Ontology embedding: a survey of methods, applications and resources. IEEE Transact Knowl Data Eng

Chen M, Wu C, Yang Z, Liu S, Chen Z, He X (2020) A multi-strategy approach for the merging of multiple taxonomies. J Inf Sci 48. https://doi.org/10.1177/0165551520952340

Cimiano P, Völker J (2005) Text2onto: a framework for ontology learning and data-driven change discovery. In: International conference on application of natural language to information systems, Springer, Heidelberg, pp 227–238, https://doi.org/10.1007/11428817_21

Costa G, Cuzzocrea A, Manco G, Ortale R (2011) Data deduplication: a review, Springer, Heidelberg, pp 385–412. https://doi.org/10.1007/978-3-642-22913-8_18

Devlin J, Chang MW, Lee K, Toutanova K (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, vol. 1 (long and short papers), Association for Computational Linguistics, Minneapolis, Minnesota, pp 4171–4186, https://doi.org/10.18653/v1/N19-1423

FakhrHosseini S, Lee C, Lee SH, Coughlin J (2024) A taxonomy of home automation: Expert perspectives on the future of smarter homes. Inf Syst Front pp 1–18, https://doi.org/10.1007/s10796-024-10496-9

Fang Y, Ryan P, Weng C (2024) Knowledge-guided generative artificial intelligence for automated taxonomy learning from drug labels. J Am Med Inform Assoc 31(9):2065–2075. https://doi.org/10.1093/JAMIA/OCAE105

Fortuna B, Mladenič D, Grobelnik M (2006) Semi-automatic construction of topic ontologies. In: Semantics, web and mining, Springer, Heidelberg, pp 121–131, https://doi.org/10.1007/11908678_8

Galárraga L, Teflioudi C, Hose K, Suchanek FM (2015) Fast rule mining in ontological knowledge bases with amie+. VLDB J 24(6):707–730. https://doi.org/10.1007/s00778-015-0394-1

Galárraga LA, Teflioudi C, Hose K, Suchanek F (2013) Amie: association rule mining under incomplete evidence in ontological knowledge bases. In: Proceedings of the 22nd international conference on world wide web, Association for Computing Machinery, New York, NY, USA, WWW '13, p 413–422, https://doi.org/10.1145/2488388.2488425

Giglou HB, D'Souza J, Auer S (2023) Llms4ol: Large language models for ontology learning. In: The semantic web - ISWC 2023 - 22nd international semantic web conference, athens, greece, november 6-10, 2023, proceedings, part I, Springer, Heidelberg, LNCS, vol 14265, pp 408–427, https://doi.org/10.1007/978-3-031-47240-4_22

Gomez JC, Moens MF (2014) A survey of automated hierarchical classification of patents. LNCS 8830:215–249. https://doi.org/10.1007/978-3-319-12511-4_11

Gyawali B, Anastasiou L, Knoth P (2020) Deduplication of scholarly documents using locality sensitive hashing and word embeddings. In: Proceedings of the 12th conference on language resources and evaluation (LREC 2020), European Language Resources Association, pp 894–903

Härtinger S, Clarke N (2016) Using patent classification to discover chemical information in a free patent database: Challenges and opportunities. J Chem Edu 93(3):534–541. https://doi.org/10.1021/acs.jchemed.5b00740

Jafari O, Maurya P, Nagarkar P, Islam KM, Crushev C (2021) A survey on locality sensitive hashing algorithms and their applications. ACM Comput Surv

Kamateri E, Salampasis M, Perez-Molina E (2024) Will AI solve the patent classification problem? World Patent Inf 78(102):294. https://doi.org/10.1016/j.wpi.2024.102294

Kang S, Agarwal S, Jin B, Lee D, Yu H, Han J (2024) Improving retrieval in theme-specific applications using a corpus topical taxonomy. In: Proceedings of the acm web conference 2024, Association for Computing Machinery, New York, NY, USA, WWW '24, p 1497–1508, https://doi.org/10.1145/3589334.3645512

Kommineni VK, König-Ries B, Samuel S (2024) Towards the automation of knowledge graph construction using large language models. In: Proceedings of the 3rd international workshop on natural language processing for knowledge graph creation co-located with 20th international conference on semantic systems (SEMANTiCS 2024), Amsterdam, The Netherlands, 17 Sept 2024, CEUR-WS.org, CEUR Workshop Proceedings, vol 3874, pp 19–34

Kozareva Z, Hovy E (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, pp 1110–1118

Kundisch D, Muntermann J, Oberländer AM, Rau D, Röglinger M, Schoormann T, Szopinski D (2022) An update for taxonomy designers: Mmethodological guidance from information systems research. Bus Inf Syst Eng 64(4):421–439. https://doi.org/10.1007/s12599-021-00723-x

Lauf F, Scheider S, Friese J, Kilz S, Radic M, Burmann A (2023) Exploring design characteristics of data trustees in healthcare – taxonomy and archetypes. In: Conference: Thirty-first european conference on information systems (ecis 2023)

Lee JS, Hsiang J (2020) Patent classification by fine-tuning BERT language model. World Patent Inf 61(1):101,965. https://doi.org/10.1016/j.wpi.2020.101965

Lee JW, Lee WK, Sohn SY (2021) Patenting trends in biometric technology of the Big Five patent offices. World Patent Inf 65(102):040. https://doi.org/10.1016/j.wpi.2021.102040

Lee K, Ippolito D, Nystrom A, Zhang C, Eck D, Callison-Burch C, Carlini N (2022) Deduplicating training data makes language models better. Proceedings of the Annual Meeting of the Association for Computational Linguistics 1:8424–8445, https://doi.org/10.18653/v1/2022.acl-long.577, arXiv: 2107.06499

Lehmann J (2009) Dl-learner: learning concepts in description logics. J Mach Learn Res 10:2639–2642

Li S, Hu J, Cui Y, Hu J (2018) DeepPatent: patent classification with convolutional neural networks and word embedding. Scientometrics 117(2):721–744. https://doi.org/10.1007/s11192-018-2905-5

Li W, Qi G, Ji Q (2020) Hybrid reasoning in knowledge graphs: combing symbolic reasoning and statistical reasoning. Semantic Web 11(1):53–62. https://doi.org/10.3233/SW-190375

Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: A robustly optimized bert pretraining approach abs/1907.11692

Maedche A, Staab S (2001) Ontology learning for the semantic web. IEEE Intell Syst 16(2):72–79. https://doi.org/10.1109/5254.920602

Martel F, Zouaq A (2021) Taxonomy extraction using knowledge graph embeddings and hierarchical clustering. In: Proceedings of the 36th annual acm symposium on applied computing, Association for Computing Machinery, New York, NY, USA, SAC '21, p 836–844, https://doi.org/10.1145/3412841.3441959

Motamedi E, Novalija I, Rei L (2024a) Classification of patents into knowledge fields: using a proposed knowledge mapping taxonomy (knowmap). In: Slovenian KDD conference, https://doi.org/10.70314/is.2024.sikdd.19

Motamedi E, Novalija I, Rei L (2024b) Taxonomy for patent classification: a step towards intelligent patent analytics. In: EKAW 2024 workshops, tutorials, posters and demos, 24th international conference on knowledge engineering and knowledge management, https://doi.org/10.13140/RG.2.2.10140.40321

Navigli R, Velardi P, Cucchiarelli A, Neri F (2004) Quantitative and qualitative evaluation of the OntoLearn ontology learning system. In: COLING 2004: Proceedings of the 20th international conference on computational linguistics, COLING, Geneva, Switzerland, pp 1043–1050

Nickerson RC, Varshney U, Muntermann J (2013) A method for taxonomy development and its application in information systems A method for taxonomy development and its application in information systems. Europ J Inf Syst 22(3):336–359. https://doi.org/10.1057/ejis.2012.26

Pena Y, Correal DE, Miranda E, González-Rojas O (2024) An innovative cost taxonomy: identifying and classifying costs of technology solutions. Int J Bus Inf Syst 45(3):397–428

Perera O, Liu J (2024) Exploring large language models for ontology learning. Issues Inf Syst 25:299–310. https://doi.org/10.48009/4_iis_2024_124

Pietrasik M, Reformat M (2020) A simple method for inducing class taxonomies in knowledge graphs. In: The semantic web, Springer International, Cham, pp 53–68, https://doi.org/10.1007/978-3-030-49461-2_4

Ristoski P, Paulheim H (2016) Rdf2vec: Rdf graph embeddings for data mining. In: The semantic web – iswc 2016: 15th international semantic web conference, kobe, japan, october 17–21, 2016, proceedings, part i, Springer, Heidelberg, p 498–514, https://doi.org/10.1007/978-3-319-46523-4_30

Sanh V, Debut L, Chaumond J, Wolf T (2019) Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. preprint 191001108

Sertkaya B (2009) Ontocomp: A protege plugin for completing owl ontologies. In: European semantic web conference, Springer, Heidelberg, pp 898–902, https://doi.org/10.1007/978-3-642-02121-3_78

Shah C, White R, Andersen R, Buscher G, Counts S, Das S, Montazer A, Manivannan S, Neville J, Rangan N, Safavi T, Suri S, Wan M, Wang L, Yang L (2025) Using large language models to generate, validate, and apply user intent taxonomies. ACM Trans Web 19(3), https://doi.org/10.1145/3732294

Silva LMVD, Köcher A, Gehlhoff F, Fay A (2024) On the use of large language models to generate capability ontologies. In: 29th IEEE international conference on emerging technologies and factory automation, ETFA 2024, Padova, Italy, 10-13 Sept 2024, IEEE, pp 1–8, https://doi.org/10.1109/ETFA61755.2024.10710775

Trattner C, Singer P, Helic D, Strohmaier M (2012) Exploring the differences and similarities between hierarchical decentralized search and human navigation in information networks. In: Acm international conference proceeding series, pp 0–7, https://doi.org/10.1145/2362456.2362474

Unterkalmsteiner M, Abdeen W (2023) A compendium and evaluation of taxonomy quality attributes. Expert Syst 40(1):e13,098, https://doi.org/10.1111/exsy.13098

Velardi P, Faralli S, Navigli R (2013) Ontolearn reloaded: A graph-based algorithm for taxonomy induction. Comput Linguist 39(3):665–707. https://doi.org/10.1162/COLI_a_00146

Wang W, Wei F, Dong L, Bao H, Yang N, Zhou M (2020) Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. Adv Neural Inf Process Syst 33:5776–5788

Weking J, Hein A (2020) A hierarchical taxonomy of business model patterns. Electron Markets 30:447–468. https://doi.org/10.1007/s12525-018-0322-5

Wilsens P, Antonio K, Claeskens G (2024) Reducing the dimensionality and granularity in hierarchical categorical variables. Springer Heidelberg. https://doi.org/10.1007/s11634-024-00614-5

Yang J, Jin H, Tang R, Han X, Feng Q, Jiang H, Zhong S, Yin B, Hu X (2024) Harnessing the power of LLMs in practice: a survey on ChatGPT and beyond. ACM Transact Knowl Discov Data 18(6), https://doi.org/10.1145/3649506

Yang Z, Luo T, Wang D, Hu Z, Gao J, Wang L (2018a) Learning to navigate for fine-grained classification. In: Proceedings of the european conference on computer vision (ECCV), pp 420–435

Yang Z, Luo T, Wang D, Hu Z, Gao J, Wang L (2018b) Learning to navigate for fine-grained classification. In: Proceedings of the european conference on computer vision (ECCV), p 438–454, https://doi.org/10.1007/978-3-030-01264-9_26