A Comparative Study of Large and Small Language Models for Domain Model Extraction

Cheng Yi Chou [0009-0009-9294-0385], Fatma Başak Aydemir [0000-0003-3833-3997], and Fabiano Dalpiaz [0000-0003-4480-3887]

Utrecht University, Utrecht, the Netherlands,
c.y.chou@students.uu.nl,{f.b.aydemir, f.dalpiaz}@uu.nl

Abstract. [*Context and Motivation*] Large language models can derive conceptual models from textual requirements, offering an off-the-shelf alternative to traditional rule-based and machine-learning-based methods. [*Question/Problem*] Comparative evidence on the validity and completeness of different large and smaller language models for the domain model derivation task remains limited. [*Principal ideas/Results*] We compare GPT-o1, Llama3-8B, and Qwen-14B with the rule-based Visual Narrator using nine datasets containing user stories and corresponding domain models. Each language model was prompted with structured templates and evaluated on class and association extraction through precision, recall, and F-scores. GPT-o1 outperformed the smaller language models and matched or exceeded Visual Narrator in most tasks. Small language models produced competitive but less consistent results, revealing efficiency–accuracy trade-offs.[*Contribution*] We provide a systematic comparison of large language models, small language models, and rule-based modeling approaches and offer an updated evaluation framework to guide future research on the balance between scale, performance, and interpretability of the automated techniques for domain model extraction.

Keywords: user stories · domain modeling · large language models · small language models · Visual Narrator

1 Introduction

Model extraction and derivation from natural language requirements have long received attention from the requirements engineering (RE) community. Analysts and researchers have sought methods to transform natural language descriptions into various types of dynamic and static conceptual models. Early work relied on rule-based techniques and linguistic heuristics [17,1], which achieved precision through explicit patterns but often failed to generalize beyond predefined contexts. Machine and deep learning pipelines [20] improved automation but required data annotation for training, limiting their applicability in practical settings.

The emergence of large language models (LLMs) offers a new opportunity to revisit this challenge. Trained on extensive corpora and capable of contextual reasoning, LLMs can interpret natural language text with greater semantic depth than earlier approaches [16]. They promise adaptability across domains, flexibility in representing diverse modeling tasks, and the potential to generate structured models directly from plain language. However, questions remain about their consistency, completeness, and alignment with human modeling practices; issues that require systematic investigation.

Recent studies have begun to examine the capabilities and limitations of LLMs in software modeling. Ferrari *et al.* [12] demonstrated that ChatGPT can generate sequence diagrams from textual descriptions but struggles with completeness and contextual precision. Chen *et al.* [7] compared prompt strategies for domain model generation, showing that examples improve accuracy, whereas chain-of-thought reasoning adds little benefit. Bragilovski *et al.* [6] found that GPT-4 can approach human recall in class identification but still exhibits distinct error profiles. While these studies highlight the potential of LLMs, few have compared them directly with small language models (SLMs) or rule-based systems using a shared evaluation framework.

Emerging work suggests that small language models (SLMs) may offer complementary advantages in software modeling tasks. Owing to their reduced parameter count and narrower training scope, SLMs can exhibit more predictable behavior and lower hallucination rates in constrained domains. Their lighter computational footprint enables cost-efficient fine-tuning and on-premise deployment, which is particularly attractive for industrial settings with privacy or resource constraints. Moreover, SLMs are often open, less bound to contractual limitations and specific deployment platforms, making them more easily integrable with rule-based or symbolic techniques, facilitating hybrid approaches that combine statistical learning with explicit domain knowledge. These characteristics position SLMs as a promising alternative for scenarios where transparency, controllability, and efficiency are critical [21,23,24].

This work addresses the lack of SLM evaluations in domain model derivation by conducting a systematic comparison of three language models—GPT-o1 as an LLM, Llama3-8B and and Qwen-14B as SLMs—against the established rule-based system Visual Narrator (VN) [17] for domain model extraction from usere stories. Our goal is to assess whether LLMs and SLMs can derive domain models that are not only syntactically valid but also semantically correct and as complete as possible. We focus on two essential elements of domain modeling (classes and associations) and evaluate model quality using established information-retrieval metrics and qualitative error analysis.

Our results reveal that GPT-o1 consistently produces models that outperform those of the SLMs and often match or exceed the rule-based baseline. The SLMs deliver competitive but less stable outcomes, revealing trade-offs between computational efficiency and modeling accuracy. These findings open new directions for research on the balance between model scale, performance, and interpretability in automated domain modeling.

The contributions of this paper are threefold. First, we introduce Visual Narrator 2.0, an LLM-capable version of the state-of-the-art rule-based domain model extraction tool Virtual Narrator. Second, we develop and integrate an additional evaluation component that refines existing model quality dimensions and supports replication and benchmarking in future studies. Third, we provide a detailed empirical evaluation across nine benchmark datasets, combining statistical and qualitative analyses. Together, these contributions advance the understanding of how LLMs can support RE tasks and lay the foundation for scalable and context-aware model extraction.

This paper is structured as follows. Sec. 2 presents the related work. Sec. 3 details our research method. Sec. 4 describes the evaluation. Sec. 5 discusses the results and Sec. 6 concludes the paper.

2 Related Work

The derivation of conceptual models from textual requirements is a long-standing strand of research [27]. Prior work has explored the automated generation of diverse artifacts such as class diagrams [8], sequence diagrams [12], and goal models [13,22] from natural language specifications. In this study, we focus specifically on domain models, which capture the key concepts and relationships within a problem space.

Arora *et al.* [1] implement a pipeline using Stanford Core NLP and GATE NLP tool kits to extract domain models from shall statements with heuristics. The pipeline extracts domain concepts, associations, generalizations, cardinalities, and attributes and is evaluated on private industrial datasets. Later, Arora *et al.* [2] use active learning to reduce superfluous entities and relations identified automated techniques. Lucassen *et al.* [19,17] introduce Visual Narrator (VN), which applies NLP and heuristic-based rules to user stories to automatically extract conceptual models, aiming to minimize human intervention while improving the interpretability of user requirements. Saini *et al.* propose DoMoBOT [20], an interactive domain model extraction tool combining rules and neural networks powered by spaCy [14] and GloVe [18] word embeddings.

Arulmohan *et al.* [3] explore how LLMs support domain model derivation from natural language text in agile product backlogs. They use GPT-3.5 in their experiments, which is outperformed by a tool implementing conditional random fields (CRF). Both perform better than VN. Chen *et al.* [7] compare the use of GPT-3.5 and GPT-4 for automated domain modeling, applying chain-of-thought prompting to capture complex domain elements. They aim to improve the precision and completeness of generated models, though the study highlights the LLMs' tendency toward omission errors and inconsistencies with best modeling practices. Bragilovski *et al.* [6,5] evaluated human analysts, a rule-based system, a machine-learning pipeline, Mistral, and GPT-4 on nearly five hundred user stories, showing that although no approach outperforms the performance or experts, LLMs perform similarly to novices.

3 Approach

This section describes the experiment design adopted to answer our research questions. Guided by Wohlin *et al.* [26], we run a comparative study that contrasts three language models with a rule-based system in the task of domain model generation.

3.1 Research Questions

Over the past few years, automated derivation of domain models from user stories has shifted from rigid heuristic pipelines to language models, but we still lack evidence on whether resource-efficient SLMs can match larger models on semantic quality, and on their error profiles. In this paper, we consider GPT-o1 as a prime example of an LLMs, Llama3-8b and Qwen-14b as examples of SLMs that can be deployed locally, and Visual Narrator as a classic rule-based approach specialized on user stories. We put forward the following research questions:

RQ1 How well do LLMs and SLMs extract domain models from user stories compared to a rule-based system?

RQ1.1 How complete are the domain models generated by GPT-o1 and the two SLMs relative to those produced by Visual Narrator?

RQ1.2 How valid are the domain models generated by GPT-o1 and the two SLMs relative to those produced by Visual Narrator?

RQ2 How do the two SLMs differ from each other and from GPT-o1 in model completeness and validity?

RQ3 How do error profiles, particularly false-positive classes and associations, differ among Visual Narrator, GPT-o1, and the two SLMs?

RQ1 quantitatively explores the performance of GPT-o1, Llama3-8b, Qwen-14b, and Visual Narrator. This work builds on Bragilovski *et al.* [6], but *i.* introduces Visual Narrator 2.0, an improved version of the state-of-the art rule based domain model extraction tool and *ii.* considers *newer GPT variants as well as two state-of-the-art SLMs*. To measure performance, we use nomenclature from conceptual modeling [10]: validity (akin to precision) and completeness (akin to recall). **RQ2** focuses specifically on a comparison across the language models, while **RQ3** is concerned with a qualitative exploration aimed at identifying patterns in the types of errors generated, which is complementary to the quantitative analysis (the latter aims at statistical testing).

3.2 Experimental Setup

Our experiment comprised four crucial elements: datasets, language models, prompt design, and evaluation metrics. We used the benchmark datasets provided by Bragilovski *et al.* [6], consisting of nine datasets. Together, there were 487 user stories, along with their corresponding domain models. The domains and brief descriptions of these datasets are summarized in Table 1. While reviewing the benchmark, we followed the guidelines of Blaha and Rumbaugh [4], and found out classes and associations for which we disagreed with the benchmarks. This can be attributed to the fact that multiple domain models may correspond to a set of requirements, depending on their purpose and the level of granularity. To accommodate for such subjectivity, we created a new version of the gold standard; this included extending it with two categories: *mandatory* and *optional*. Mandatory elements are explicitly stated in the user stories, while optional ones are not directly mentioned but can be inferred from domain knowledge. The optional elements are counted as true positives if identified, but are not treated as false negatives if not identified following the guidelines of Blaha and Rumbaugh [4]. Table 1 summarizes the gold domain models prior and after our revision.

We selected language models that receive regular updates, are easily accessible through an open-source platform[1], and can follow natural language instructions through

[1] https://huggingface.co/

Table 1: Datasets Descriptions and Metrics: US = number of user stories; C_{old} and C_{new} are the number of classes in the old and new gold standard; A_{old} and A_{new} are the number of associations.

Dataset	**Description**	**US**	$\mathbf{C}_{old}$	$\mathbf{C}_{new}$	$\mathbf{A}_{old}$	$\mathbf{A}_{new}$
Camperplus	A camp management system for admins, parents, and counselors to track activities and share documents.	55	17	19	23	26
Fish&Chips	A restaurant system supporting takeaway, delivery, and improving outdated kitchen workflows.	50	9	9	8	8
Grocery	A grocery chain's internal system for HR, scheduling, payroll, and employee self-service.	49	9	10	8	12
Planningpoker	An agile estimation tool supporting collaborative story point voting and backlog refinement.	53	6	6	6	6
Recycling	A waste management system for organizing recycling types, locations, schedules, and tracking actions.	51	11	10	11	10
School	A school management system for grades, attendance, messaging, and home-based learning.	61	17	18	23	24
Sports	A CRM system for fitness centers supporting lesson booking, trainer management, and registration.	63	13	13	13	12
Supermarket	An online grocery platform offering delivery, in-store navigation, and personalized promotions.	51	11	12	13	16
Ticket	An event ticketing platform with user profiles, guest checkout, and a resale marketplace.	54	10	10	13	13
Total		**487**	**103**	**107**	**118**	**127**

prompting. We initially evaluated the performance of multiple recent models using the interactive chat interface on Hugging Face, as it allowed for a quick qualitative assessment of model outputs. Models that produced repetitive or unintelligible answers were excluded. The final set of models included GPT-o1 (a reasoning-focused, instruction-tuned model with an unspecified parameter size), Llama3-8B Instruct[2] (an instruction-tuned variant with 8 billion parameters from Meta's Llama series), and DeepSeek-R1-Distill-Qwen-14B[3] (a 14-billion-parameter distilled model from the Qwen series, optimized through knowledge distillation for improved inference efficiency).

We focused on prompt design, as this can influence model performance. Following [25], we adopted three prompting strategies: *i.* persona, *ii.* template pattern, and *iii.* chain of thought. The models were assigned the role of a RE expert and provided with an output template to facilitate processing. We structured prompts into two steps for GPT-o1 and four steps for SLMs, based on observations from our pilot experiment. GPT-o1 could process all instructions, handle the full list of user stories, and follow the instructions accurately to generate the output. However, separating the user stories from the instructions slightly improved its performance. In contrast, SLMs generally failed to

[2] https://huggingface.co/meta-llama/Meta-Llama3-8B-Instruct

[3] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-14B

follow all instructions when presented at once, often forgetting earlier parts. Therefore, the prompts were divided into four parts for SLMs, with each step delivered separately.

Our task involved two main stages: *i.* identifying relevant classes and *ii.* determining their associations. Figure 1 illustrates these processes. Although similar, the two tasks differ in their inputs and outputs. For class identification, the input is user stories and instructions, and the output is a set of classes. For association identification, the input includes gold-standard classes, user stories, and instructions, and the output is a set of associations. The process follows Blaha and Rumbaugh's guidelines [4]: candidate classes and associations are first identified, and then irrelevant ones are excluded based on exclusion criteria. Both code and datasets are available online[4].

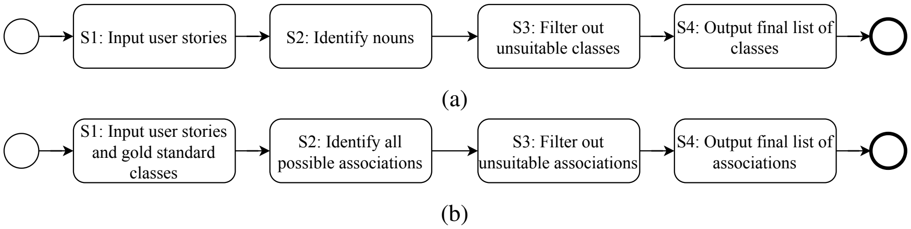
S1: Input user stories
S2: Identify nouns
S3: Filter out unsuitable classes
S4: Output final list of classes
(a)
S1: Input user stories and gold standard classes
S2: Identify all possible associations
S3: Filter out unsuitable associations
S4: Output final list of associations
(b)

Fig. 1: Steps for extracting (a) classes and (b) associations.

3.3 Evaluation Design

Starting with the quantitative evaluation, we used $F_{0.5}$, F_1 and F_2 scores because they combine precision and recall in a single and interpretable number. $F_{0.5}$ gives more weight to precision while still considering recall, and serves as an indicator of model validity. In contrast, F_2 emphasizes recall and shows the completeness of the model outputs. F_1 provides a balanced point.

Since language model outputs are stochastic, we run of GPT-o1 five times and Llama3-8B and Qwen-14B ten times each, with the differing numbers reflecting budget constraints. By contrast, predictions of VN are deterministic for it is rule based, so a single run per dataset is sufficient. Performance varied across rounds for the LLMs and SLMs, potentially due to random variation rather than systematic effects. Therefore, it was necessary to evaluate whether the observed performance differences among models were statistically significant. Given the small number of datasets in this study and that the independent variable comprised more than two categories, we used the Friedman test to assess whether there is a statistically significant difference among the models. If significant differences were found, the Nemenyi post-hoc test was applied to perform pairwise comparisons and identify differences in performance [9].

In addition to the quantitative evaluation, we conducted a qualitative analysis to gain deeper insights into the models' performance. Following the approach of Bragilovski

[4] https://github.com/rexchou0715/VisualNarrator-v2

et al. [6], we considered seven false positive categories for classes (Table 2a) and three for associations (Table 2b), derived from Blaha and Rumbaugh [4]. To ensure reproducibility, we constructed decision trees for taggers to guide the classification of false positives for both classes and associations, which are available in the online appendix. Certain branches of the trees were adapted from [6], and our main addition was a branch designed to handle hallucinated terms as well as verb and noun forms.

Table 2: False Positive Categories, primarily based on [4].

False Positive Category	**Description**
Irrelevant	A class that has little or no connection to the problem domain.
Implementation	A class that is used for system realization rather than for representing real-world entities.
Operation	A class that represents an operation applied to objects rather than an entity in its own right.
Redundant	A class that represents a concept already expressed by another class; the less descriptive one should be removed.
Vague	A class that is too general or lacks sufficient specificity to represent a distinct concept.
Role	A class that represents a temporary or external role rather than an intrinsic part of the domain.
Attribute	A class whose name primarily describes properties of individual objects rather than independent entities.

(a) FP categories for classes

Irrelevant	An association that lies outside the problem domain.
Implementation	An association that reflects implementation details rather than domain relationships.
Redundant/Derived	An association that can be inferred or defined through other existing associations.

(b) FP categories for associations

4 Evaluation

Since our research involves conducting multiple rounds of experiments and evaluating their outcomes, a substantial amount of manual effort was initially required. To address this, we developed Virtual Narrator 2.0 (in our appendix) that has two main capabilities: *i.* LLM integration for domain model generation and *ii.* automated evaluation of the generated outputs by comparing them with the given golden standard.

The evaluation process considers not only literal matches between elements but also their semantic similarity. Because language models often produce varied expressions even under instructions, we observed that some outputs correctly refer to an element but use a different term. In such cases, we avoid penalizing the models by creating a synonym dictionary that records all equivalent terms generated during the experiments by the following process. We first let the LLMs and SLMs generate candidate classes (Step 1) through Visual Narrator 2.0. Next, the first author reviewed all false positive and unmatched classes from the gold standard (Step 2) and flagged all false positives that might refer to the same real-world concepts as those in the unmatched list (Step 3). Then, the first author checked the user stories to verify whether each false positive referred to the same concept as the one in the unmatched list (Step 4). The first and the third authors discussed the flagged classes that might refer to the same real-world entities (Step 5) and added the class labels to the synonym dictionary if they agree on their similarity (Step 6). The dictionary has a total of 158 synonyms for 60 classes.

We consider two variants of the original, rule-based VN in line with previous research [5]: VN Precision-Oriented (VN-P) and VN Recall-Oriented (VN-R), both of which are part of the model comparison. VN-P serves as the baseline for RQ1.1, and VN-R for RQ1.2. The difference between these versions is in the threshold that is set for including a noun or a noun phrase as a class in the output (lower threshold for VN-R). Only the classes extracted by VN are considered in the evaluation, as its association extraction does not follow the same guidelines used in our study. Including associations would therefore introduce inconsistencies and reduce the validity of the comparison.

4.1 Quantitative Analysis

Table 3 compares the five approaches for extracting domain models from user stories (GPT-o1, Llama3-8B, Qwen-14B, VN-P, VN-R) on class identification.

Table 3: $F_{0.5}$, F_1, and F_2 scores for class identification, highlighting the best results in yellow and the second best in gray.

Dataset	**GPT-o1**			**Llama3-8B**			**Qwen-14B**			**VN-P**			**VN-R**		
	$F_{0.5}$	F_1	F_2	$F_{0.5}$	F_1	F_2	$F_{0.5}$	F_1	F_2	$F_{0.5}$	F_1	F_2	$F_{0.5}$	F_1	F_2
Recycling	0.423	0.453	0.490	0.256	0.318	0.428	0.234	0.274	0.348	0.385	0.286	0.227	0.320	0.370	0.439
Supermarket	0.680	0.764	0.873	0.482	0.499	0.532	0.558	0.632	0.745	0.597	0.640	0.689	0.597	0.640	0.689
Planningpoker	0.643	0.700	0.773	0.365	0.396	0.463	0.437	0.510	0.624	0.682	0.600	0.536	0.588	0.615	0.645
Camperplus	0.791	0.747	0.708	0.601	0.520	0.468	0.658	0.676	0.703	0.672	0.581	0.512	0.479	0.571	0.708
Grocery	0.631	0.688	0.761	0.532	0.564	0.612	0.471	0.545	0.667	0.682	0.462	0.349	0.564	0.608	0.660
Sports	0.689	0.682	0.680	0.562	0.567	0.578	0.579	0.620	0.680	0.536	0.375	0.288	0.379	0.480	0.652
Ticket	0.637	0.648	0.662	0.539	0.545	0.568	0.304	0.356	0.439	0.595	0.556	0.521	0.595	0.556	0.521
School	0.625	0.638	0.654	0.563	0.531	0.515	0.601	0.658	0.739	0.561	0.579	0.598	0.556	0.623	0.707
Fish&Chips	0.613	0.649	0.691	0.410	0.467	0.546	0.387	0.456	0.565	0.476	0.333	0.256	0.217	0.298	0.473
Macro Avg.	0.637	0.663	0.699	0.479	0.490	0.523	0.470	0.525	0.612	0.576	0.490	0.442	0.477	0.529	0.610
Macro SD.	0.097	0.090	0.104	0.113	0.084	0.061	0.143	0.140	0.138	0.100	0.130	0.165	0.140	0.122	0.104

Looking at validity ($F_{0.5}$), GPT-o1 outperforms both SLMs across all datasets and exceeds VN-P in almost all cases, with Planningpoker and Grocery as the main exceptions (VN-P slightly higher). VN-P also surpasses the SLMs in most datasets. For completeness (F_2), GPT-o1 dominates Llama3-8B, with Qwen-14B and VN-R exhibiting very similar performance (0.612 vs. 0.610) and being the runner-up of GPT-o1 on most datasets. It is remarkable how, although the variants of VN are from a decade ago, they are still up-to-part (if not better) than the examined SLMs.

Statistical tests ($\alpha = 0.05$) confirm these differences. The Friedman test shows significance for both metrics (validity: $p = 0.0001246$; completeness: $p = 0.003283$). Post-hoc (Nemenyi) and Cohen's d effect size tests confirm that GPT-o1 significantly outperforms Llama3-8B ($p = 0.0029$, $d = 2.343$) and Qwen-14B ($p = 0.0056$, $d = 1.929$) in validity, but not VN-P ($p = 0.3543$, $d = 0.842$). For completeness, GPT-o1 only shows a significant advantage over Llama3-8B ($p = 0.0015$, $d = 1.786$), with no significant differences from Qwen-14B ($p = 0.2208$, $d = 0.913$), or VN-R ($p = 0.1532$, $d = 0.997$).

Figure 2 visualizes the distribution of scores of each dataset across different runs using box plots. Note that VN is excluded since its results are deterministic. GPT-o1 achieves higher and more stable scores overall, with performance gaps narrowing for completeness. Qwen-14B becomes more competitive when completeness is prioritized, whereas Llama3-8B remains the least consistent. This is yet another signal of the superiority of GPT-o1 over the two examined SLMs for the task at hand.

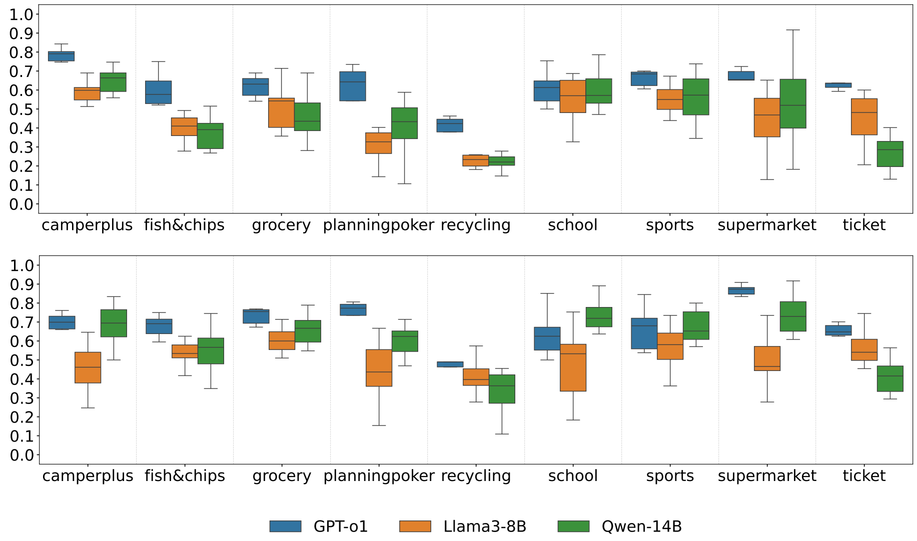

Fig. 2: Class $F_{0.5}$ and F_2 score distribution by model and dataset.

Table 4 reports the $\mathrm{F}_{0.5}$, F_1 and F_2 scores for association extraction across all datasets and models. Since we are only comparing three alternatives, unlike Table 3 where we had five contenders, we highlight the best but not the runner-up.

Table 4: $\mathrm{F}_{0.5}$, F_1, and F_2 scores for identifying associations.

Dataset	**GPT-o1**			**Llama3-8B**			**Qwen-14B**		
	$\mathrm{F}_{0.5}$	F_1	F_2	$\mathrm{F}_{0.5}$	F_1	F_2	$\mathrm{F}_{0.5}$	F_1	F_2
Recycling	0.525	0.578	0.647	0.469	0.512	0.580	0.426	0.494	0.596
Supermarket	0.929	0.925	0.920	0.728	0.720	0.718	0.718	0.721	0.728
Planningpoker	0.799	0.832	0.871	0.744	0.736	0.741	0.727	0.752	0.779
Camperplus	0.731	0.671	0.621	0.536	0.446	0.385	0.573	0.564	0.557
Grocery	0.738	0.809	0.896	0.580	0.590	0.615	0.556	0.621	0.706
Sports	0.722	0.789	0.870	0.579	0.587	0.598	0.531	0.574	0.640
Ticket	0.877	0.918	0.965	0.618	0.564	0.523	0.718	0.725	0.739
School	0.681	0.682	0.682	0.382	0.358	0.346	0.423	0.472	0.540
Fish&Chips	0.781	0.833	0.896	0.552	0.564	0.590	0.641	0.684	0.738
Macro Avg.	0.754	0.782	0.819	0.576	0.564	0.566	0.590	0.623	0.669
Macro SD.	0.116	0.117	0.130	0.114	0.119	0.133	0.119	0.104	0.088

Looking at validity ($\mathrm{F}_{0.5}$), GPT-o1 consistently outperforms both SLMs across all datasets, with scores ranging from 0.525 (Recycling) to 0.929 (Supermarket) and an average of 0.754. The margins vary by dataset, being smallest in Planningpoker and largest in Supermarket and School. Between the SLMs, there is no clear dominance, as Llama3-8B leads in five datasets while Qwen-14B performs better in four, occasionally achieving larger margins (e.g., Fish&Chips, Ticket).

For completeness (F_2), GPT-o1 again demonstrates the strongest performance, with scores between 0.647 and 0.965 (avg. 0.819). It remains above 0.85 in six datasets, indicating strong coverage of relevant associations. Qwen-14B follows, averaging 0.669, and consistently outperforms Llama3-8B (avg. 0.566).

Statistical tests ($\alpha = 0.05$) confirm the significant differences. The Friedman test shows significance for both metrics (validity: $p = 0.0012$)completeness (F_2): $p = 0.0001$, $W = 1$). The post-hoc (Nemenyi) tests further show that GPT-o1 significantly outperforms both Llama3-8B ($p = 0.0062$, $d = 2.113$) and Qwen-14B ($p = 0.0028$, $d = 2.898$) in validity. For completeness, GPT-o1 is significantly better than Llama3-8B ($p = 0.0000656$, $d = 2.267$), while its dominance over Qwen-14B is not statistically significant ($p = 0.0855$, $d = 2.221$), likely due to the limited sample size.

Figure 3 presents the $\mathrm{F}_{0.5}$ and F_2 scores for each dataset, showing the variation across different runs of LLMs and small SLMs in the association extraction task. GPT-o1 exhibits higher medians and narrower interquartile ranges, particularly in Supermarket, Camperplus, and Fish&Chips. Llama3-8B shows lower medians and greater variability, while Qwen-14B generally lies between the two, occasionally approaching GPT-o1's performance (e.g., Planningpoker). The patterns for completeness are similar, with GPT-o1 maintaining the highest stability and the differences among models narrowing slightly in Recycling.

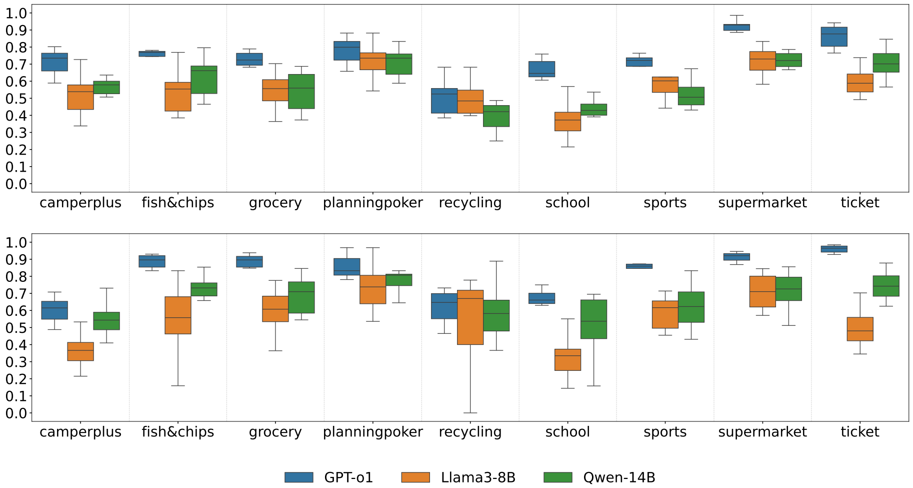

Fig. 3: $F_{0.5}$ (top) and F_2 (bottom) score distribution by model and dataset for association extraction.

4.2 Qualitative Analysis

This section examines false positive patterns in class and association identification. Figure 4 compares the class false positive profiles of GPT-o1, Llama3-8B, Qwen-14B, VN-P, and VN-R. For each dataset, we aggregate all experimental rounds, compute the proportion of each error type, and summarize their distributions using box plots. This visualization highlights error tendencies rather than absolute counts.

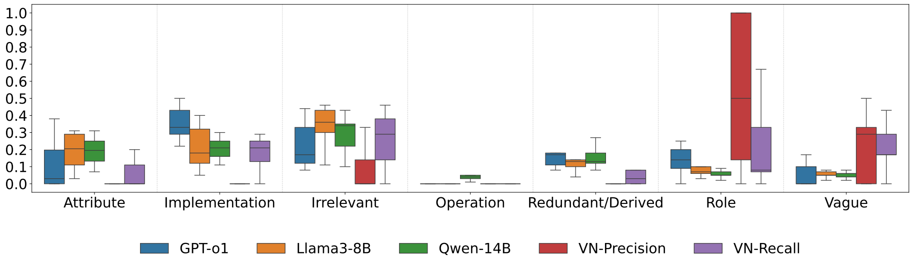

Fig. 4: Distribution of class false positives by error types and models.

VN and language models show distinct error patterns. VN-P produces a narrow range of mistakes, mainly *Role*, *Vague*, and *Irrelevant* errors, while VN-R shows greater variety (because of the lower threshold for class inclusion) but similar patterns. Both variants frequently make *Vague* errors due to difficulty in handling complex nouns (VN uses a classical NLP pipeline). For example, in the Ticket dataset, the system identifies "information" as a class, although it is unclear whether this refers to "artist information" or "event information". They also tend to make *Role* errors by extracting system-oriented entities, such as "admin", as standalone classes. The roles appear frequently in the "As a" part of user story templates, leading VN to assign them higher weights. Occasionally, VN also produces *Irrelevant* errors by identifying relative pronouns, such as "which" or "where," as potential classes, likely due to how sentence chunking.

In contrast, language models distribute their errors more evenly and show a stronger tendency toward *Redundant/Derived* classes, a category that VN rarely triggers. For instance, in the School dataset, Qwen-14B produced both "grade" and "school grade" within the same experiment round. Among the LLMs, GPT-o1 makes fewer *Attribute* and *Irrelevant* errors, whereas both SLMs more frequently confuse attributes with classes or generate out-of-scope entities. GPT-o1, however, shows a slightly higher rate of *Role* and *Implementation* errors. For the remaining categories (*Vague*, *Redundant/Derived*, *Operation*), all three models behave similarly. The *Operation* category shows low error rates across all models because language models can effectively distinguish nouns from verbs. Qwen-14B exhibits a slightly higher rate due to its tendency to over-generate elements and extract compound words. For instance, in Camperplus, it identified "consent form submission" as a class; however, the term emphasizes "submission", which is an operation rather than a domain entity.

Figure 5 compares the association false positive profiles of GPT-o1, Llama3-8B, and Qwen-14B. Note that VN is excluded from this part of the evaluation, as explained at the beginning of Section 4. *Redundant/Derived* errors are the most frequent across all types. GPT-o1 shows the highest median and widest interquartile range, indicating that it sometimes adds links already implied, for example "Student–Assignment," which can be inferred from "Student–Class" and "Class–Assignment." *Implementation* errors are rare for all models, with GPT-o1 showing the fewest. In the gold standard class list, there is no reference to classes related to the development of a system, *i.e.* technical classes. Any reference to technical classes in the generated output indicates that the model has ignored the instructions provided in the prompt and hallucinated. This phenomenon is especially evident in the SLMs. *Irrelevant* errors occur moderately across models, often from invented or loosely related entities. GPT-o1 has the lowest median but wider spread, while the SLMs display higher medians with less variation.

5 Discussion and Limitations

5.1 Discussion

Our study demonstrates that the GPT-o1 LLM consistently achieved higher validity and completeness than both SLMs and the rule-based Visual Narrator. This result suggests that scale, the training data, and advanced contextual reasoning significantly improve the accuracy of class and association identification. The SLMs delivered performance

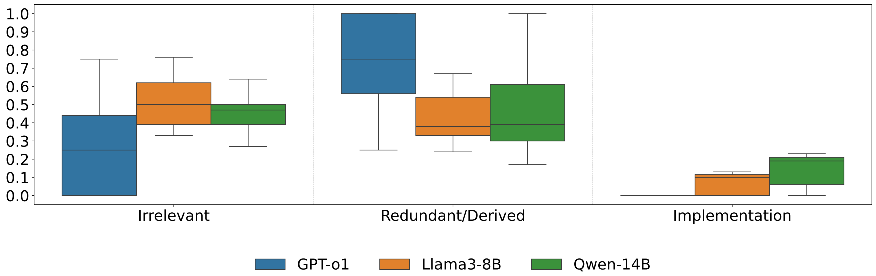

Fig. 5: Distribution of false positives by error types and models for associations.

on the level of the VN; while the SLMs are better suited than the LLMs for educational and resource-constrained environments, the even lighter rule-based VN is still in the same league, if not better, for many datasets.

The first and the third authors manually studied all results and improved the gold standard as well. The analysis revealed recurring error patterns across models. Larger models tended to generalize associations correctly but occasionally introduced spurious relationships, while smaller models often omitted valid elements or failed to interpret domain-specific terms. These differences highlight that model scale influences not only accuracy but also the types of reasoning errors produced. VN, on the other hand, is fully interpretable and the identified issues could be fixed programmatically.

Our results raise questions about how LLMs reason about software concepts. Despite their generally good performance, they still encounter difficulties with domain-specific semantics, showing that human oversight remains essential. This observation aligns with prior work emphasizing the need for hybrid pipelines [11] that combine LLM automation with manual SE activities. Further research should investigate how prompting strategies, fine-tuning, and feedback loops can balance automation and reliability in conceptual modeling tasks.

Across datasets, our analysis also reveals differences in task difficulty. Some domains appear inherently more challenging for both humans and machines. For example, the Recycling dataset yielded consistently lower scores across all models, mirroring prior observations that even human analysts struggled to construct coherent domain models from its user stories [5]. In contrast, the Supermarket dataset achieved the highest scores, and human participants in the same study also performed well on this domain. These results suggest that dataset characteristics–such as clarity, vocabulary consistency, and conceptual regularity–strongly influence the quality of both human and automated model derivation.

Implications for practice. Large and small language models demonstrate potential to accelerate early modeling activities by extracting preliminary conceptual structures directly from textual requirements, reducing manual effort and without requiring specialized solutions (e.g., classical NLP pipelines). Organizations can integrate such models into existing tool chains to support analysts in identifying classes and associations. However, industrial adoption requires attention to data confidentiality, explainability, and validation workflows to ensure that generated models remain trustworthy and compliant with organizational standards. In such cases, SLMs have an advantage over LLMs for resource consumption, ease of on-premise deployment, and more predictable behavior. With appropriate governance and human oversight, LLM-based model extraction can support faster prototyping, better stakeholder communication, and more consistent documentation across projects. Yet, the effects on humans' cognition need to be investigated, as recent studies [15] have shown how the use of ChatGPT leads to a so called 'cognitive debt', i.e., lower neural connectivity in people's brain.

5.2 Threats to Validity

Following the guidelines by Wohlin *et al.* [26], we consider potential threats to construct, internal, external, and conclusion validity.

Construct validity may be affected by the operationalization of conceptual modeling quality, as our metrics focus on classes and associations while omitting attributes, specializations, and behavioral aspects. This is a conscious choice: classes and associations are at the basis of domain models, and other elements (like attributes) are dependent on classes' existence.

Conclusion validity concerns the interpretation of quantitative differences; to address this, we employed statistical tests and complemented numerical analysis with qualitative inspection. Error profiling is rare in NLP4RE research, and we advocate it as as step to go beyond the raw numbers and statistical values. Nevertheless, we have not tested the effectiveness of the generated models in action, i.e., how well they support requirements analysts or developers.

Internal validity may be influenced by prompt design and random variations in LLM outputs; we mitigated this risk through standardized prompting and multiple runs per model. We were interested in the models' performance with their predefined settings, although an alternative would have been controlling temperature and seed. However, this would not have been possible for GPT-o1.

External validity is limited by the benchmark datasets, which reflect a specific set of user stories and may not represent all industrial domains or modeling styles. To reduce bias, we have re-tagged the dataset and included mandatory and optional elements, plus a set of synonyms. Nevertheless, replication with larger datasets, additional modeling tasks, and alternative evaluation frameworks will be necessary to strengthen the generalizability of our findings.

6 Conclusions

Our main contribution is a systematic comparison of LLMs and SLMs for conceptual model extraction from natural language requirements. We evaluated three models (GPT-o1, Llama3-8B, and Qwen-14B) against the rule-based Visual Narrator across nine benchmark datasets. The findings confirm that GPT-o1 performs consistently better than the other models and often matches or exceeds the rule-based baseline in both precision and recall. The SLMs demonstrate competitive performance, thereby serving as practical alternatives where computational efficiency is a priority; yet, they do not outperform the lightweight, NLP-based Visual Narrator.

Our research identifies several avenues for future work. We plan to extend the analysis to additional modeling elements such as attributes, multiplicities, and specific relations. We aim to refine the evaluation framework by integrating human judgment metrics and exploring automated semantic alignment techniques. LLM-as-a-judge evaluation setup is also an interesting direction to explore. Finally, we will examine how the insights from this comparison can inform the design of mixed-initiative systems that integrate LLMs into RE workflows responsibly and transparently.

Data availability The replication package of this work is available at https://github.com/rex-chou0715/VisualNarrator-v2

References

1. Arora, C., Sabetzadeh, M., Briand, L., Zimmer, F.: Extracting domain models from natural-language requirements: approach and industrial evaluation. In: International Conference on Model Driven Engineering Languages and Systems. pp. 250–260 (2016)
2. Arora, C., Sabetzadeh, M., Nejati, S., Briand, L.: An active learning approach for improving the accuracy of automated domain model extraction. Transactions on Software Engineering and Methodology **28**(1), 1–34 (2019)
3. Arulmohan, S., Meurs, M.J., Mosser, S.: Extracting domain models from textual requirements in the era of large language models. In: International Conference on Model Driven Engineering Languages and Systems Companion. pp. 580–587. IEEE (2023)
4. Blaha, M., Rumbaugh, J.: Object-Oriented Modeling and Design with UML, 2/E. Pearson Education India (2007)
5. Bragilovski, M., van Can, A.T., Dalpiaz, F., Sturm, A.: Leveraging machines to derive domain models from user stories. Requirements Engineering pp. 1–23 (2025)
6. Bragilovski, M., Van Can, A.T., Dalpiaz, F., Sturm, A.: Deriving domain models from user stories: Human vs. machines. In: International Requirements Engineering Conference. pp. 31–42. IEEE (2024)
7. Chen, K., Yang, Y., Chen, B., López, J.A.H., Mussbacher, G., Varró, D.: Automated domain modeling with large language models: A comparative study. In: International Conference on Model Driven Engineering Languages and Systems. pp. 162–172. IEEE (2023)
8. Deeptimahanti, D.K., Sanyal, R.: Semi-automatic generation of UML models from natural language requirements. In: India Software Engineering Conference. pp. 165–174 (2011)
9. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine learning research **7**(Jan), 1–30 (2006)
10. España, S., Ruiz, M., González, A.: Systematic derivation of conceptual models from requirements models: a controlled experiment. In: International Conference on Research Challenges in Information Science. pp. 1–12. IEEE (2012)
11. Fan, A., Gokkaya, B., Harman, M., Lyubarskiy, M., Sengupta, S., Yoo, S., Zhang, J.M.: Large language models for software engineering: Survey and open problems. In: International Conference on Software Engineering: Future of Software Engineering. pp. 31–53. IEEE (2023)
12. Ferrari, A., Abualhaijal, S., Arora, C.: Model generation with llms: From requirements to UML sequence diagrams. In: Model-Driven Requirements Engineering Workshop. pp. 291–300. IEEE (2024)
13. Güneş, T., Aydemir, F.B.: Automated goal model extraction from user stories using nlp. In: International Requirements Engineering Conference. pp. 382–387. IEEE (2020)
14. Honnibal, M., Montani, I., Van Landeghem, S., Boyd, A.: spaCy: Industrial-strength Natural Language Processing in Python (2020). https://doi.org/10.5281/zenodo.1212303

15. Kosmyna, N., Hauptmann, E., Yuan, Y.T., Situ, J., Liao, X.H., Beresnitzky, A.V., Braunstein, I., Maes, P.: Your brain on ChatGPT: Accumulation of cognitive debt when using an AI assistant for essay writing task. arXiv preprint arXiv:2506.08872 **4** (2025)
16. Kumar, P.: Large language models (LLMs): survey, technical frameworks, and future challenges. Artificial Intelligence Review **57**(10), 260 (2024)
17. Lucassen, G., Robeer, M., Dalpiaz, F., Van Der Werf, J.M.E., Brinkkemper, S.: Extracting conceptual models from user stories with Visual Narrator. Requirements Engineering **22**, 339–358 (2017)
18. Pennington, J., Socher, R., Manning, C.: GloVe: Global vectors for word representation. In: Conference on Empirical Methods in Natural Language Processing. pp. 1532–1543. Association for Computational Linguistics (2014)
19. Robeer, M., Lucassen, G., Van Der Werf, J.M.E., Dalpiaz, F., Brinkkemper, S.: Automated extraction of conceptual models from user stories via NLP. In: International Requirements Engineering Conference. pp. 196–205. IEEE (2016)
20. Saini, R., Mussbacher, G., Guo, J.L., Kienzle, J.: DoMoBOT: An AI-empowered bot for automated and interactive domain modelling. In: International Conference on Model Driven Engineering Languages and Systems Companion. pp. 595–599. IEEE (2021)
21. Schick, T., Schütze, H.: Its not just size that matters: Small language models are also few-shot learners. In: Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: Human language technologies. pp. 2339–2352 (2021)
22. Sharfuddin, A., Breaux, T.: Generative goal modeling. In: International Requirements Engineering Conference. pp. 92–103. IEEE (2025)
23. Van Nguyen, C., Shen, X., Aponte, R., Xia, Y., Basu, S., Hu, Z., Chen, J., Parmar, M., Kunapuli, S., Barrow, J., et al.: A survey on small language models. In: Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing-Natural Language Processing in the Generative AI Era. pp. 807–821 (2025)
24. Wang, F., Zhang, Z., Zhang, X., Wu, Z., Mo, T., Lu, Q., Wang, W., Li, R., Xu, J., Tang, X., et al.: A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with llms, and trustworthiness. ACM Transactions on Intelligent Systems and Technology **16**(6), 1–87 (2025)
25. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., Schmidt, D.C.: A prompt pattern catalog to enhance prompt engineering with ChatGPT. arXiv preprint arXiv:2302.11382 (2023)
26. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., et al.: Experimentation in software engineering, vol. 236. Springer (2012)
27. Yue, T., Briand, L.C., Labiche, Y.: A systematic review of transformation approaches between user requirements and analysis models. Requirements Engineering **16**(2), 75–99 (2011)