Subevent Structure as a Predictor of Entity Identity Change in Procedural Text

Kyeongmin Rim, James Pustejovsky
Brandeis University
Waltham, MA, USA
krim@brandeis.edu, jamesp@brandeis.edu

Abstract

We test whether the subevent structure encoded in VerbNet-GL predicts entity identity change in procedural text, using only the verb's lexical specification and no training data. From each VN verb class's SEMANTICS block we extract an aspectual classification and an I/O count, yielding a predicted *dynamic event topology* (DET). On the observation side, ten large language models (LLMs) annotate per-entity *dynamic object mode* (DOM) labels over ∼3100 OpenPI steps, from which we derive observed DET for comparison. The VN-only predictor achieves 67.4% precision (F1 = 0.35) for transformation, showing that formal subevent structure carries genuine predictive signal only for the most common topology, but low performance for other topologies. For events VN predicts as having no result state, only 24% are confirmed as no-change by silver, indicating that the remaining outcomes arise from the argument side of the composition. These results provide empirical evidence that event semantics is distributed across predicate and argument: the VN supplies the subeventual skeleton, but is not sufficient to determine the final outcome.

Keywords: VerbNet, entity state tracking, event semantics, procedural text, distributed compositionality, co-composition

1. Introduction

A central claim of Generative Lexicon (GL) theory (Pustejovsky, 1995) is that semantic composition is not a static process of function application, but a generative one involving *co-composition*. Under this mechanism, the verb's event interpretation is sensitive to the *qualia structure* of its arguments, a structured representation encoding how an object comes into being (AGENTIVE), what it is for (TELIC), its formal type (FORMAL), and its constitutive parts (CONSTITUTIVE). In the classic example of *bake*, the event resolves to transformation for a *potato* (a natural kind lacking an AGENTIVE quale) but shifts to emergence for a *cake* (an artifact whose AGENTIVE quale licenses a creation reading). To test this linguistic theory, procedural text provides an ideal test case, as it describes the *lifecycle of entities* through sequences of actions that transform their state (e.g., flour → dough → bread), and datasets such as OpenPI (Tandon et al., 2020) provide per-step state annotations that record these empirical outcomes.

VerbNet (VN; Kipper et al., 2008) organizes ∼300 verb classes by shared syntactic and semantic behavior. Its extension with GL subevent semantics (hereafter VN-GL; Brown et al., 2019) decomposes each class into temporally ordered subevents ($e_1, e_2, \ldots$), uniquely equipping it to serve as a formal basis for this analysis. In the GL framework, qualia roles act as a representational overlay on this subevent structure, specifying the predicative constraints and oppositions associated with each phase. This subeventual skeleton is designed to capture precisely the kind of entity change that OpenPI annotates: each event can be modeled as an input/output (I/O) transformation (Rim et al., 2023) where participants enter a process and exit in a potentially altered state, even when the result is not textually mentioned (Ježek and Melloni, 2011).

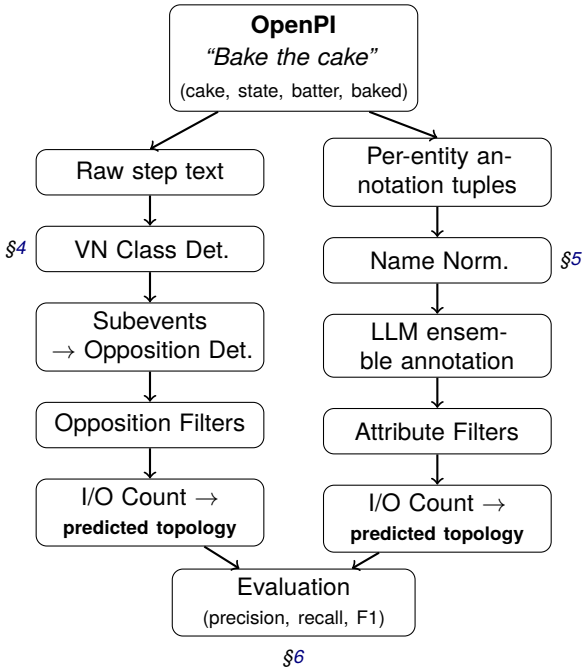

Figure 1: Evaluation Design: Two independent paths derive step-level DET from the same OpenPI data, compared at alignment.

Proceedings of the Workshop on Structured Linguistic Data and Evaluation, pages 52–63

While VN-GL subevent structure defines the structural constraints of such transformations, we argue that the verb's lexical specification remains inherently underspecified. Thus, the final interpretation, including the identity of the resulting state, must emerge from a mutual specification between the verb and its participants. This suggests that event meaning is effectively *distributed* across the predicate and its arguments, rather than being concentrated within the verb alone.

In this paper, we present the first empirical evaluation of this distributed interpretation by testing the predictive reach of the verb's formal contribution (its subevent structure) in isolation. We ask: how much of the observed entity change in procedural text is predicted by the verb alone? This verb-only baseline constitutes a controlled ablation of argument information, directly measuring the "compositional gap" that co-composition is intended to fill.

The answer matters because systematic failures in a verb-only model reveal exactly where argument qualia is required, providing empirical evidence for GL's distributed interpretation. Moreover, a symbolic resource offers falsifiable, inspectable predictions and a finite scope for targeted extension.

We define a step-level *dynamic event topology* (DET) that characterizes the structural shape of entity identity change at each procedural step, and derive it via two independent paths: (1) *predicted* DET, extracted symbolically from VN-GL subevent structure (aspectual classification and I/O count), requiring no training data; and (2) *observed* DET, derived from per-entity *dynamic object mode* (DOM) labels that ten LLMs assign using OpenPI's crowdsourced attribute tuples, with no VN-GL information in the prompt. Comparing the two measures how much of the observed entity change the verb's lexical specification alone can account for. Our results show that while verb-level semantics reliably predicts transformation in transition events, it systematically fails in three cases: low emergence precision, unmodeled state changes in process events, and argument-qualia overrides in destruction-class verbs. These are precisely the points where GL predicts that argument-driven co-composition is required to determine the event outcome, providing empirical evidence that event semantics is distributionally composed across predicate and argument structure.

Our contributions are:

1. A symbolic method for deriving dynamic event topologies from VN-GL subevent structure (aspectual classification + I/O count), requiring no training data.
2. A silver-standard dynamic object mode (DOM) dataset over OpenPI, produced by majority vote across ten LLMs and assessed against manual gold annotation.
3. An empirical evaluation of VN-GL's predictive reach, with error categories grounded in GL's co-composition theory.

2. Related Work

2.1. Entity State Tracking in Procedural Text

Entity state tracking focuses on monitoring how participants change through sequences of actions. ProPara (Dalvi et al., 2018) introduced the task with a closed vocabulary (location and existence) over scientific process paragraphs. The OpenPI dataset (Tandon et al., 2020) scaled it to open-vocabulary state-change tuples ($\sim$30000) over WikiHow procedures, where each tuple records an entity, an attribute, and its before/after values (e.g., *onion*, *state*, *whole*, *diced*). Quality issues in OpenPI's original crowdsourcing have been addressed by filtering unreliable state changes (Wu et al., 2023) and canonicalizing entity and attribute strings (Zhang et al., 2024).

Neural entity trackers have progressed from structured conditional random field (CRF) models (Gupta and Durrett, 2019b) through transformer-based approaches (Gupta and Durrett, 2019a) to interactive entity networks (Tang et al., 2020). On the knowledge-based side, ProComp (Clark et al., 2018) and Lexis (Kazeminejad et al., 2021) predict state changes symbolically from VerbNet rules without training data, while PROPOLIS (Faghihi et al., 2023) integrates symbolic semantic parsing with neural components. More recently, EvEntS ReaLM (Spiliopoulou et al., 2022) tackles entity state reasoning via language models. Our work differs from all of these in that we do not predict entity states directly; instead, we test whether a formal lexical resource can predict the event's change topology (DET) from verb semantics alone.

2.2. Formal Models of Subevent Structure

Event structure theories decompose verbs into temporal phases. Vendler's (1967) four-way aspectual classification (state, activity, accomplishment, achievement) was enriched by Moens and Steedman (1988) with transitions and culmination points, providing a direct predecessor to Pustejovsky's subevent decomposition. Pustejovsky (2013) formalizes this dynamic event model, which was then implemented in VerbNet (Kipper et al., 2008), built on Levin's (1993) diathesis-based verb classification, by Brown et al. (2018), later refined in Brown et al. (2019) and Brown et al. (2022). The resulting VN-GL framework utilizes subevent variables and predicate opposition to model phase transitions.

Tu et al. (2024) adapt this opposition structure to build graph representations augmenting Abstract Meaning Representation (AMR) with subevent annotations. Fine-grained intra-class semantic distinctions have been documented by Kazeminejad et al. (2022).

Our work relies on GL's theory of co-composition (Pustejovsky, 1995), which holds that arguments actively contribute to an event's interpretation. We adopt the Dynamic Argument Structure (DAS) taxonomy (Jezek, 2017) and the Generalized Result Role thesis (Ježek and Melloni, 2011) to map verb-level subevents to the entity-level changes observed in procedural data. Crucially, resources like VN-GL localize this structure in the verb class, while we argue that event semantics is distributionally composed across both predicate and arguments.

2.3. Silver Data Adjudication and Distillation

Generating high-quality annotations for open-vocabulary state tracking has proven difficult via crowdsourcing (Tandon et al., 2020; Wu et al., 2023). Recent work has shifted toward using Large Language Models (LLMs) as high-fidelity meta-annotators, even outperforming human crowd workers in structured semantic tasks (Gilardi et al., 2023). Following the Models-Vote Prompting paradigm (Oniani et al., 2023), ensemble methods are increasingly used to mitigate family-specific hallucinations and biases (Kamen and Kamen, 2025). Our work builds on (Rim and Pustejovsky, 2026), which establishes that the relationship between state outcomes and semantic structures is systematic and recoverable. However, while typical silver-labeling pipelines stop at model consensus, we introduce a layer of *theory-informed post-processing* (Veselovsky et al., 2023): GL's distinction between extrinsic change (location, possession) and intrinsic identity change overrides model consensus for entities whose annotations involve only spatial or ownership attributes.

3. Task Definition

3.1. Dynamic Object Mode

On the observation side, each OpenPI entity, represented by its (entity, attribute, before, after) tuples, receives a label that we call the *dynamic object mode* (DOM): the mode of identity change that the entity undergoes at a given step. The term reflects the argument side of the distributed composition: the verb (predicate) determines the structural shape of the event (DET), while the DOM records what happens to each participant. Step-level DET is then derived mechanically from these per-entity DOM labels (Section 5.4), enabling comparison against the VN-GL predicted DET.

We ground the DOM taxonomy in the Dynamic Argument Structure (DAS) framework of Jezek (2017), which characterizes five modes for event participants: *initiation* (brought into existence), *termination* (ceases to exist), *modification* (an attribute value changes), *transfer* (information moves through a medium), and *persistence* (present but unaffected). We narrow DAS *modification* to type-level change ($T_1 \rightarrow T_2$): an entity is **transformed** only if it becomes a different kind of thing (e.g. batter $\rightarrow$ cake). Attribute-level changes (location, temperature) that preserve the entity's type fall under **persistent**. The DOM taxonomy is:

- **Created**: the entity comes into being at this step.
- **Terminated**: the entity ceases to exist at this step.
- **Transformed**: the entity persists with identity preserved but undergoes a type-level change ($T_1 \rightarrow T_2$).
- **Persistent**: the entity is present and participated but not meaningfully affected.

A fifth label, **bystander**, flags entities whose OpenPI tuples are too noisy for meaningful classification (e.g., coreference errors, abstract concepts, annotator artifacts); these are excluded from downstream DET derivation and all evaluation metrics.

3.2. Dynamic Event Topology

Where DOM describes what happens to each entity, the *dynamic event topology* (DET) describes the structural shape of the event as a whole, the predicate side of the distributed composition.

DET is grounded in two theoretical foundations in Generative Lexicon (GL). First, GL's aspectual classification (Pustejovsky, 1995) groups Vendler's (1967) four classes into three: *states* (static, no change), *processes* (activities, no result state), and *transitions* (accomplishments and achievements, which encode a result state). Only transitions receive a DET assignment; since stative verbs rarely appear as main predicates in procedural text, only the transition/process distinction is operative here. Second, for transitions, DET is derived from the predicative opposition that GL's FORMAL quale encodes: by counting which participant roles appear across the opposition pairs in the subevent structure, we obtain the number of entities entering (n_{in}) versus exiting (n_{out}) the result state. This yields five topologies:

- **Transformation** ($1 \rightarrow 1$): one entity enters and one exits with altered type.

- **Emergence** ($0 \rightarrow n$): no input entity; one or more new ones appear in the result state.
- **Dissolution** ($n \rightarrow 0$): one or more entities enter but none persists in the result.
- **Convergence** ($n \rightarrow 1,\ n \geq 2$): multiple inputs merge into a single output.
- **Divergence** ($1 \rightarrow n,\ n \geq 2$): a single input divides into multiple outputs.

Process and state events receive no DET assignment.

3.3. Evaluation Design

The core evaluation compares *predicted* DET, derived symbolically from VN-GL subevent structure (Section 4.2), against *observed* DET, derived mechanically from per-entity DOM labels annotated by an LLM ensemble over OpenPI steps (Section 5.4). The two derivations are fully independent: the prediction side uses no OpenPI annotations, and the observation side uses no VN-GL information (Figure 1). Their alignment measures how much of the observed entity change the verb's lexical specification alone can account for.

4. Predicted DET from VN-GL

4.1. Motivation for a Resource-Based Approach

An obvious alternative is to skip VN-GL entirely and ask an LLM to predict entity state changes directly. This would likely achieve higher accuracy. But this work is not a state-tracking benchmark; it is a test of a linguistic formalism.

A formal resource provides what an LLM cannot. First, *falsifiability*: VN-GL makes a specific, inspectable claim per verb class; when a class mispredicts, the failure traces to a specific formal structure (a missing result subevent, an unmodeled co-patient), not to opaque model weights. Second, *controlled ablation*: the verb-only DET baseline is an intentional removal of qualia information from the argument side, directly testing GL's distributed co-compositionality thesis: how much event meaning resides in the predicate versus the argument? An LLM conflates both, having seen verb-argument combinations in training, and cannot isolate the predicate's contribution. Third, *resource auditing*: VN-GL is a finite, curated resource ($\sim$300 classes, $\sim$6000 lemmas); understanding its predictive reach provides a concrete roadmap for targeted extension.

4.2. DET Prediction Pipeline

Given a sentence from OpenPI dataset, the prediction pipeline derives a DET of the events in the sentence through three stages. No OpenPI entity or state annotation is used; the prediction is entirely verb-determined.

Stage 1: VerbNet class assignment The VerbNet Parser (VNP; Gung and Palmer, 2021) maps the step's main verb to a VerbNet class and extracts surface thematic roles from the syntactic frame. From the VN 3.4 XML for the assigned class, we select the frame whose declared syntax roles best match VNP's surface roles (ties favor the earlier frame, as VN orders frames from canonical to specialized).

Stage 2: Opposition pair detection and aspectual gating From the selected frame's SEMANTICS block, each `PRED` element is extracted with its subevent variable, polarity (`bool="!"` for negated), thematic role arguments, and any `PredSpecific` arguments. VN-GL uses three event variable types (Brown et al., 2022): `E` (overall event variable, holds throughout the entire event); and two temporally ordered subevent types: $\ddot{e}_n$ (process subevents, unbounded activity) and e_n (state-transition subevents). We consider only predicates at plain e_n variables that have at least one ThemRole argument, skipping `E` and $\ddot{e}$ predicates.

These predicates are grouped by (predicate name, ThemRole). Within each group, we look for *opposition pairs*: the same predicate on the same role at two different subevent indices, with evidence of opposition. Two patterns are recognized:[1]

- **Pattern 1** (negation): opposite polarity at different indices (e.g., $\neg$`cooked(`e_1`, Patient)` and `cooked(`e_4`, Patient)`);
- **Pattern 2** (PredSpecific): different `PredSpecific` values at different indices (e.g., `has_val(`e_1`, Patient, Initial_State)` and `has_val(`e_3`, Patient, Result)`).

For each pair, the entry at the earlier index is the *initial side*; the later is the *result side*.

If at least one opposition pair is found, the event is a *transition*; otherwise it is a *process* and receives no DET assignment. This gatekeeper is deliberately conservative: the entity-change rate

[1] Across all 1602 frames in VN 3.4, these two patterns account for all detected opposition structure (1225 negation-based pairs across 37 predicates; 21 PredSpecific-based pairs across 3 predicates). 692 additional cases of the same predicate and role at different indices with same polarity and no PredSpecific difference are continuity tracking, not oppositions. Two predicates (`created`, `exist`) appear in the VN 3.4 XML but are absent from the published predicate inventory of Brown et al. (2022); the published inventory lists `degraded` where the XML uses `degradation_material_integrity`.

observed in process events (Section 6.2) measures the compositional gap that argument qualia must fill.

Stage 3: I/O role counting and existence classification Not all detected opposition pairs constitute type-level identity change. In GL, location change is extrinsic (Pustejovsky, 1995): it alters a relation between the entity and an external domain, not the entity's intrinsic type; possession transfer similarly changes ownership without altering the object itself. Pairs on these predicates encode spatial or ownership changes that preserve entity identity. These pairs still contribute to the aspectual gate (they confirm the event has subevent structure), but their roles are excluded from I/O counting.[2]

From the remaining identity-relevant pairs, we count distinct roles: n_{in} from initial-side entries whose role is in the input set (Patient, Theme, Co-Patient, Co-Theme, Material, Asset), and n_{out} from result-side entries in the output set (Patient, Theme, Result, Product). When the class declares a Co-Patient or Co-Theme in its THEMROLES, n_{in} is raised to at least 2.

Raw counting treats every role as both input and output if it appears on both sides of a pair. However, some oppositions describe an entity coming into or going out of existence, rather than persisting through a state change. For such roles, counting them on both sides is incorrect: an entity that ceases to exist is input-only (should not count toward n_{out}), and an entity that comes into existence is output-only (should not count toward n_{in}).

To identify these cases, we manually tagged each of the 163 VN 3.4 predicates (Brown et al., 2022) with its *existence semantics*: *existence* predicates (8) whose positive form entails the entity is present, *cessation* predicates (4) whose positive form entails it is no longer present, and all others with no existence implication. [3]

For each negation-based pair (Pattern 1), the predicate's tag and polarity at each side determine the per-role adjustment:

- entity ceases to exist (e.g., `alive` → ¬`alive`, or ¬`destroyed` → `destroyed`): the role is removed from n_{out};
- entity comes into existence (e.g., ¬`created` → `created`): the role is removed from n_{in};
- all other Pattern 1 pairs and all Pattern 2 pairs: no adjustment.

The raw and adjusted $(n_{\text{in}}, n_{\text{out}})$ determines the predicted DET from the topology definitions in Section 3.2. Namely, we evaluate the existence semantics rule as an ablation in Section 6.2.

For example, the SEMANTICS block for COOKING-45.3 contains ¬`cooked(`e_1`, Patient)` and `cooked(`e_4`, Patient)`. These form a negation-based opposition pair on Patient. Since `cooked` has no existence implication, no adjustment is applied: Patient contributes to both n_{in} and n_{out}, yielding $(1, 1) \rightarrow$ transformation.

5. Observed DET from Silver DOM

5.1. OpenPI Overview

OpenPI (Tandon et al., 2020) provides ∼30000 open-vocabulary state-change tuples over 810 WikiHow documents. Each tuple records an entity, an attribute, and its before/after values at a given procedural step (e.g., *onion*, *state*, *whole*, *diced*). The dataset covers six WikiHow categories: Food & Entertaining, Home & Garden, Hobbies & Crafts, Cars & Other Vehicles, Sports & Fitness, and a residual category.

Raw OpenPI tuples record attribute-level state changes (e.g., temperature, location, shape) but do not directly encode DOM, which abstracts over these attributes to characterize the entity's biographical mode of identity change (*created*, *terminated*, *transformed*, or *persistent*). The same entity may appear with multiple attribute tuples at a single step (e.g., *onion*: state whole→diced, location cutting-board→pan), and surface entity strings are unnormalized ("the deep fryer" and "deep fryer" are separate entries). Quality issues in the original crowdsourcing have been documented by Wu et al. (2023), who found ∼32% of state changes unreliable, and by Zhang et al. (2024), who attempted entity canonicalization with limited success.

5.2. Silver DOM Annotation

We derive silver-quality DOM labels via LLM ensemble annotation, with post-processing informed by GL's distinction between extrinsic and intrinsic change.

Annotation distillation For each step, models receive the document goal, step text, and the set of raw (attribute, before, after) tuples for each entity.

[2] This exclusion is symmetric with the observation side, where OpenPI entities whose attribute tuples consist solely of location or possession changes are relabeled to **persistent** (Section 5.2).

[3] Location: `has_location`, `has_orientation`, `has_position`. Possession: `has_possession`, `has_information`. Existence (positive = entity present): `alive`, `appear`, `be`, `created`, `exist`, `give_birth`, `procreate`, `visible`. Cessation (positive = entity gone): `destroyed`, `disappear`, `suffocated`, `voided`. Classification based on predicate definitions in Appendix A and semantic clusters in Appendix C of Brown et al. (2022).

	Annotator	Docs	Steps	Entities	cre	ter	tra	per	bys
	Majority vote	**639**	**3101**	**14642**	**4.8**	**2.1**	**37.3**	**40.7**	**15.1**
Silver (per-model)	Gemma-2B	*(same as above)*		8682	55.1	4.5	23.3	15.5	1.7
	Gemma-7B			14423	8.0	3.5	66.7	11.9	9.9
	Gemma-3-12B			14596	5.7	3.8	56.0	23.1	11.3
	Qwen2.5-14B			14543	10.3	2.5	69.0	4.0	14.2
	Qwen3-14B			14588	6.2	3.5	74.4	11.2	4.6
	Haiku-4.5			14638	4.9	7.9	28.3	38.7	20.2
	Sonnet-4.6			14621	4.0	6.4	18.5	47.3	23.8
	GPT-4.1-mini			14606	8.2	5.3	24.6	40.7	21.2
	GPT-5-mini			14611	3.0	3.5	17.9	59.5	16.1
	GPT-5.2			14614	3.7	2.2	14.6	51.3	28.1
Gold		20	104	532	2.4	1.5	56.2	21.2	18.6

Table 1: DOM label distribution (%) for silver (per-model and majority vote) and gold annotations, after post-processing. Entity counts vary across models because each model may fail to parse or label a subset of entities. Column headers: cre = created, ter = terminated, tra = transformed, per = persistent, bys = bystander.

They are instructed to adjudicate these noisy, low-level strings into the DOM taxonomy (Section 3.1); the full prompt and a few-shot example are given in Appendix A.[4] Crucially, no VN-GL information or subevent definitions are included in the prompt; the LLMs classify entity change solely from the OpenPI tuples, ensuring that the observation side is fully independent of the prediction side. This treats the LLM as a meta-annotator, distilling clean, high-level semantic features from crowdsourced strings (Gilardi et al., 2023).

Ensemble adjudication Following the Models-Vote Prompting paradigm (Oniani et al., 2023), we employ ten instruction-tuned LLMs: five local GPU models (Gemma-2B, Gemma-7B, Gemma-3-12B (Gemma Team et al., 2024), Qwen2.5-14B, Qwen3-14B (Qwen Team, 2025)) and five commercial APIs (GPT-4.1-mini, GPT-5-mini, GPT-5.2, Claude-Haiku-4.5, Claude-Sonnet-4.6). This "wisdom of the crowd" mitigates family-specific hallucinations and biases inherent in single-model pipelines (Kamen and Kamen, 2025). The final silver label for each entity is determined by plurality vote across the ensemble; ties are broken by a fixed priority ordering that favors more informative modes (transformed $>$ created $>$ terminated $>$ persistent).

Pre-processing: entity normalization OpenPI entity mentions are free-form text annotations with no standardized form. We apply minimal normalization (lowercasing, stripping articles, collapsing whitespace), which merges "the deep fryer" with "deep fryer" but not "deep fryer" with "fryer." Zhang et al. (2024) attempted LLM-based entity canonicalization via GPT-3.5-turbo with 3-shot prompting, but achieved an entity clustering F1 of only 0.495, meaning roughly half of coreference links were missed. Given this limited reliability, we opt for the simpler deterministic normalization and treat residual duplicates as separate entities.

Post-processing: non-identity-change override As discussed in Section 4.2, location and possession changes do not constitute type-level identity change. On the observation side, we force-relabel any entity whose OpenPI annotations consist solely of location-related attributes (`location`, `orientation`, `position`, `placement`, `movement`, `motion`, `direction`, `distance`) or possession-related attributes (`ownership`, `possession`) to **persistent**, regardless of model consensus. This affects $\sim$21% of entities (3112 of 14642).

Silver dataset statistics The silver dataset comprises 639 documents (all training-split documents with at least one entity tuple), 3101 steps, and 14642 entities. Table 1 shows the per-model and majority-vote label distributions. Two model clusters emerge: transformation-heavy models (Qwen, Gemma-7B/12B; 56–74%) versus conservative models (Claude, GPT; 15–28%), with the majority vote mediating at 37%/41% (transformed/persistent). The non-identity-change override (Section 5.2) relabels 3112 entities (21.3%) to persistent.

5.3. Human Verification

5.3.1. Concreteness Filtering

Not all OpenPI procedures describe physical transformations. To focus human verification on documents where entity identity change is expected, we

[4]The prompt uses simplified label names (e.g., "transform" for **transformed**, "destroyed" for **terminated**) that were established before the final taxonomy was formalized; we verified that models interpret these consistently across all ten LLM annotators.

implement a two-layer concreteness filter. Layer 1 uses lexico-syntactic heuristics: a keyword blocklist for video game titles, spaCy (Honnibal et al., 2023) named entity recognition (NER) for WORK_OF_ART entities, and exclusion of goals whose root verb is abstract (*be*, *seem*, *involve*). Layer 2 computes mean noun concreteness via the Brysbaert et al. (2014) norms ($\sim$40k lemmas, 1–5 scale), normalized to $[0, 1]$ and thresholded at > 0.8. This identifies 422 of 639 documents as concrete. The filter is also used as a post-hoc robustness check on the full evaluation (Section 6.2).

5.3.2. Gold Annotation

We randomly sampled 20 documents from the 422 concrete-filtered set; however, four (20%) were discarded and redrawn after manual inspection revealed non-physical procedures that passed the automatic filter (e.g., clothing selection, exercise routines), indicating that the concreteness heuristic is a coarse screen rather than a reliable classifier. All steps within each document were annotated, yielding 104 steps and 532 entities (Table 1, bottom row).

Annotation guidelines include a decision flowchart, edge-case policies, and worked examples; the full text will be released with the dataset. Two graduate students independently labeled the sample using the five-way DOM taxonomy. Inter-annotator agreement was moderate (Cohen's $\kappa = 0.41$; $\kappa = 0.48$ excluding bystander), largely depressed by noise in the underlying OpenPI tuples, as many borderline entities receive inconsistent attribute annotations in the source data, making DOM assignment genuinely ambiguous. The two sets were adjudicated into a single gold standard used for all evaluations in Section 6.

5.4. Observed DET Derivation

Step-level DET is derived mechanically from per-entity DOM counts; it is not a separate annotation. If any entity is **transformed**, or exactly one is **terminated** and one **created**, the step receives **transformation**. **Created**-only steps yield **emergence**; **terminated**-only steps yield **dissolution**. When multiple entities are **terminated** and at least one is **created**, the step is **convergence**; the reverse yields **divergence**. Steps where all entities are **persistent** receive no DET label.

6. Evaluation

6.1. Silver Label Quality

Table 3 compares each model's DOM labels against the adjudicated gold standard (532 entities, **bystander** excluded). The models fall into two clusters. Transformation-heavy models (Qwen3-14B, Qwen2.5-14B, Gemma-7B) reach 0.63–0.68 entity-level accuracy with strong **transformed** F1 (0.74–0.81) but near-zero **persistent** F1. Conservative models (Claude, GPT) balance **transformed** and **persistent** F1 (0.48–0.65 and 0.50–0.56) at lower overall accuracy. The majority vote mediates these clusters, achieving the highest step-level DET accuracy (0.654) with balanced per-label F1, consistent with the complementary-bias findings of Oniani et al. (2023).

Table 2 confirms this cluster structure via pairwise Cohen's κ at the entity level. Intra-cluster agreement is moderate to substantial (transformation-heavy: κ = 0.48–0.60; conservative: κ = 0.55–0.63), while inter-cluster agreement is lower (κ = 0.21–0.44). Gemma-2B is an outlier with uniformly low agreement (κ = 0.16–0.30). This diversity supports the ensemble design.

6.2. DET Prediction Evaluation

Table 4 compares verb-only DET predictions (Section 4.2) against the silver DET derived from majority-vote DOM labels. Of 2940 overlapping steps, 863 (29.4%) agree on the DET label.

Among transition events, **transformation** $(1{\to}1)$ achieves 67.4% precision over 596 steps: when VN-GL predicts a single-participant transition, the silver observation agrees in roughly two out of three cases, without any training data or argument information.[5] The overall accuracy (29.4%) falls below a majority-class baseline that always predicts transformation (58.8%). This deficit is driven by two factors: process events ($n = 1655$), which receive no DET assignment and only 24% of which are confirmed as no-change by silver, and convergence over-prediction (388 steps at 0% precision) from Co-role declarations in THEMROLES. However, where the predictor commits to transformation, it outperforms the baseline (67.4% vs. 58.8%), and the non-transformation predictions provide structurally informative error categories that a majority-class baseline cannot.

The process-event gap has two sources. First, VNP may misassign a process class to a verb that VN-GL actually encodes as a transition (a parser error that could be addressed with improved class disambiguation). Second, some verbs are genuinely aspectually unbounded but still produce observable state changes on their arguments in procedural context (*stir the batter* changes the batter's state despite *stir* being a process verb); this is precisely

[5] Restricting to the 422 concrete-filtered documents (Section 5.3.1) yields 70.0% transformation precision over 404 steps (27.5% overall, 1946 steps), suggesting that non-physical procedures introduce modest noise but do not materially affect the findings.

	Model	1	2	3	4	5	6	7	8	9	10
1.	Gemma-2B	—									
2.	Gemma-7B	.30	—								
3.	Gemma-3-12B	.25	.48	—							
4.	Qwen2.5-14B	.28	.51	.51	—						
5.	Qwen3-14B	.28	.56	.55	.60	—					
6.	Haiku-4.5	.22	.33	.44	.39	.36	—				
7.	Sonnet-4.6	.17	.25	.36	.30	.26	.63	—			
8.	GPT-4.1-mini	.23	.29	.40	.36	.32	.62	.57	—		
9.	GPT-5-mini	.16	.21	.32	.26	.23	.55	.57	.58	—	
10.	GPT-5.2	.16	.20	.30	.27	.23	.56	.61	.57	.61	—

Table 2: Pairwise entity-level Cohen's κ across the ten silver annotator models. Mean κ = 0.38.

Model	N	κ	DOM	DET
Gemma-2B	241	.026	.261	.183
Gemma-7B	377	.122	.647	.538
Gemma-3-12B	408	.105	.598	.510
Qwen2.5-14B	396	.088	.634	.538
Qwen3-14B	423	.190	.683	.587
Haiku-4.5	400	.275	.570	.500
Sonnet-4.6	387	.180	.478	.433
GPT-4.1-mini	400	.190	.505	.433
GPT-5-mini	415	.171	.489	.500
GPT-5.2	382	.164	.474	.452
Majority	**417**	**.219**	**.585**	**.654**

Table 3: Individual model and majority-vote DOM labels evaluated against adjudicated gold (bystander excluded). N = entities with labels from model and gold overlap; κ = Cohen's kappa; DOM = entity-level accuracy; DET = step-level accuracy.

the case where GL predicts that the argument's qualia must supply the result interpretation that the verb alone does not encode.

6.3. Discussion

6.3.1. Error Sources

VerbNet Parser accuracy All DET predictions depend on VNP (Gung and Palmer, 2021) correctly assigning a VerbNet class. VNP was trained on declarative sentences, but OpenPI procedures are imperative (e.g., *Dice the onion*). We use the Stanza dependency parser (Qi et al., 2020) to detect imperative sentences and prepend "you" before parsing (*you dice the onion*), converting imperative to pseudo-declarative form. VNP accuracy on this augmented input was not measured; misassigned classes propagate directly into wrong DET predictions.

OpenPI noise Crowdsourced tuples contain ∼32% unreliable annotations (Wu et al., 2023), surface entity strings are unnormalized, and coreference is unresolved. This noise affects both gold inter-annotator agreement (IAA; $\kappa = 0.41$) and the silver quality ceiling, as models must adjudicate noisy tuples into clean DOM labels. The two behavioral clusters in Table 3 reflect genuine ambiguity in the source data as much as model disagreement.

Cardinality extraction Convergence overprediction is the largest single error source: 388 steps predicted as convergence ($n_{\text{in}} \geq 2$), none matching silver. This arises from the Co-role adjustment: classes declaring Co-Patient or Co-Theme in THEMROLES receive $n_{\text{in}} \geq 2$, but in procedural text these classes rarely produce actual convergence events. Dissolution predictions (28 steps, 32.1% precision) are recovered by the existence classification (Section 4.2), but silver observes transformation for 11/28 of these steps, indicating that argument qualia overrides the predicate's encoded termination in procedural context. Divergence (73 steps at 0% precision) is a further error source where VN-GL structural cardinality does not match the observed topology.

6.3.2. Distributed Compositionality

The transformation precision (67.4% over 596 steps) shows that the verb's subevent structure carries substantial predictive information for transition events, without access to any argument information. Three systematic failure modes indicate where the predicate alone is insufficient.

VN-GL predicts emergence for 175 steps, with 51 matching silver (29.1% precision), while silver observes 468 emergence steps overall. Emergence in procedural text is typically implicit (*flour + water → dough*), where the result entity's existence depends on the argument's qualia structure, specifically the AGENTIVE quale that licenses a creation reading (Pustejovsky, 1995). The existence classification detects predicates like `created` and `exist` and removes the role from n_{in}, but this captures only a fraction of actual emergence events.

Only 24% of steps classified as process events are confirmed as no-change by silver. Verbs such as *stir*, *spread*, and *rub* are aspectually unbounded but produce observable state changes on their arguments in procedural context. The verb's aspectual class yields no DET prediction; the argument's participation must force a result interpretation.

Of 28 dissolution predictions, 9 match silver (32.1% precision). These are cases where VN-GL encodes entity termination via cessation pred-

	Raw				+ Exist. class.			
DET	**N**	**P**	**R**	**F1**	**N**	**P**	**R**	**F1**
transformation	684	.658	.260	.373	596	.674	.233	.346
emergence	115	.348	.085	.137	175	.291	.109	.159
dissolution	0	–	–	–	28	.321	.043	.077
convergence	388	.000	.000	.000	388	.000	.000	.000
divergence	73	.000	.000	.000	73	.000	.000	.000
no DET	1680	.239	.762	.364	1680	.239	.762	.364
acc.		**.303**				**.294**		
baseline		*.588*				*.588*		

Table 4: DET prediction alignment with ablation. *Raw*: opposition pairs detected, I/O roles counted directly. + *Exist. class.*: existence/cessation tags applied to adjust cardinality for emergence and dissolution. "no DET" includes both process events ($n = 1655$) and transitions with $(0, 0)$ cardinality ($n = 25$). Baseline: always predict transformation.

icates (`destroyed`, `suffocated`), and the existence classification correctly identifies the role as input-only. Where dissolution is mispredicted, silver typically observes transformation, indicating that argument qualia overrides the predicate's encoded termination in procedural context.

Together, these failure modes delineate the boundary between what the predicate contributes and what the argument must supply, consistent with GL's claim that event meaning is distributed across predicate and argument structure.

6.4. Future Work

The cardinality-based DET heuristic could be replaced with a neural classifier trained over SEMANTICS blocks, predicting DET directly from predicate structure rather than relying on role-counting rules. On the argument side, integrating noun-level semantic resources (e.g., WordNet hypernym chains, qualia structure annotations (Pustejovsky et al., 2009)) could extend the symbolic pipeline to capture the argument qualia that verb-only prediction misses, testing whether VN-GL combined with a noun resource improves DET prediction without training data. A neuro-symbolic encoder for GL qualia structure could learn dense representations of predicate-argument interactions, combining the interpretability of formal resources with the coverage of distributional models.

7. Conclusion

We evaluated VerbNet-GL subevent structure as a symbolic predictor of entity identity change in procedural text. From VN-GL SEMANTICS blocks, we derived aspectual classifications and I/O counts to produce symbolic DET predictions over ∼3100 OpenPI steps, evaluated against silver DOM labels from a ten-model LLM ensemble. The verb-only predictor achieves 67.4% precision for transformation over 596 steps (29.4% overall), indicating that formal subevent structure carries predictive signal for transition events. Systematic failures—low emergence precision, process-event state changes, and argument-qualia overrides in destruction-class verbs—correspond to cases where GL theory predicts that argument qualia is required to complete the event interpretation, providing evidence that event semantics is distributed across predicate and argument structure. The VN-DET code, gold and silver datasets, and annotation guidelines will be released upon publication.

8. Limitations

Our evaluation depends on the VerbNet Parser for class assignment, which was trained on declarative sentences and has not been evaluated on imperative procedural text; the pseudo-declarative conversion (prepending "you") is a heuristic mitigation whose accuracy remains unmeasured, and misassigned classes propagate directly into DET prediction errors. The silver standard, while validated against gold annotation, inherits noise from OpenPI's crowdsourced tuples (∼32% unreliable per Wu et al. (2023)). Among the 2077 mispredictions, convergence over-prediction (388 steps from Co-role declarations), divergence over-prediction (73 steps), and process events (only 24% of 1655 steps confirmed as no-change by silver) are the most prominent residual error sources. Finally, the evaluation covers only WikiHow procedures, which skew toward physical tasks; generalization to other procedural domains (e.g., scientific protocols with different verb distributions, or assembly manuals relying more on nominal predicates) remains untested.

9. Bibliographical References

Susan Windisch Brown, Julia Bonn, James Gung, Annie Zaenen, James Pustejovsky, and Martha Palmer. 2019. VerbNet Representations: Subevent Semantics for Transfer Verbs. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 154–163, Florence, Italy. Association for Computational Linguistics.

Susan Windisch Brown, Julia Bonn, Ghazaleh Kazeminejad, Annie Zaenen, James Pustejovsky, and Martha Palmer. 2022. Semantic Representations for NLP Using VerbNet and the Generative Lexicon. *Frontiers in Artificial Intelligence*, 5.

Susan Windisch Brown, James Pustejovsky, Annie Zaenen, and Martha Palmer. 2018. Integrating Generative Lexicon Event Structures into VerbNet. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods*, 46(3):904–911.

Peter Clark, Bhavana Dalvi, and Niket Tandon. 2018. What Happened? Leveraging VerbNet to Predict the Effects of Actions in Procedural Text.

Bhavana Dalvi, Lifu Huang, Niket Tandon, Wentau Yih, and Peter Clark. 2018. Tracking State Changes in Procedural Text: A Challenge Dataset and Models for Process Paragraph Comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1595–1604, New Orleans, Louisiana. Association for Computational Linguistics.

Hossein Rajaby Faghihi, Parisa Kordjamshidi, Choh Man Teng, and James Allen. 2023. The Role of Semantic Parsing in Understanding Procedural Text.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

James Gung and Martha Palmer. 2021. Predicate Representations and Polysemy in VerbNet Semantic Parsing. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 51–62, Groningen, The Netherlands (online). Association for Computational Linguistics.

Aditya Gupta and Greg Durrett. 2019a. Effective use of transformer networks for entity tracking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Aditya Gupta and Greg Durrett. 2019b. Tracking discrete and continuous entity state for process understanding. In *Proceedings of the Third Workshop on Structured Prediction for NLP*. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, and Henning Peters. 2023. spaCy: Industrial-strength Natural Language Processing in Python. Zenodo.

Elisabetta Jezek. 2017. Dynamic Argument Structure. *Linguistic Issues in Language Technology*, 15(3).

Elisabetta Ježek and Chiara Melloni. 2011. Nominals, polysemy, and co-predication. *Journal of Cognitive Science*, 12(1):1–31.

Ariel Kamen and Yakov Kamen. 2025. Majority rules: LLM ensemble is a winning approach for content categorization. *arXiv preprint arXiv:2511.15714*.

Ghazaleh Kazeminejad, Martha Palmer, Susan Windisch Brown, and James Pustejovsky. 2022. Componential analysis of English verbs. volume 5. Frontiers.

Ghazaleh Kazeminejad, Martha Palmer, Tao Li, and Vivek Srikumar. 2021. Automatic Entity State Annotation using the VerbNet Semantic Parser. In *Proceedings of the Joint 15th Linguistic Annotation Workshop (LAW) and 3rd Designing Meaning Representations (DMR) Workshop*, pages 123–132, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A large-scale classification of English verbs. *Language Resources and Evaluation*, 42(1):21–40.

Beth Levin. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press.

Marc Moens and Mark Steedman. 1988. Temporal ontology and temporal reference. *Computational Linguistics*, 14(2):15–28.

David Oniani, Jordan Hilsman, Hang Dong, Shiven Verma, Fengyi Gao, and Yanshan Wang. 2023. Large language models vote: Prompting for rare disease identification. *arXiv preprint arXiv:2308.12890*.

James Pustejovsky. 1995. *The Generative Lexicon*. MIT Press.

James Pustejovsky. 2013. Dynamic Event Structure and Habitat Theory. In *Proceedings of the 6th International Conference on Generative Approaches to the Lexicon (GL2013)*, pages 1–10, Pisa, Italy. Association for Computational Linguistics.

James Pustejovsky, Jessica Moszkowicz, Olga Batiukova, and Anna Rumshisky. 2009. GLML: Annotating Argument Selection and Coercion. In *Proceedings of the Eight International Conference on Computational Semantics*, pages 169–180, Tilburg, The Netherlands. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108. Association for Computational Linguistics.

Qwen Team. 2025. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Kyeongmin Rim and James Pustejovsky. 2026. Missing links: LLM-augmentation of event triggers of state changes in the OpenPI dataset. In *Proceedings of the 2026 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2026)*. European Language Resources Association.

Kyeongmin Rim, Jingxuan Tu, Bingyang Ye, Marc Verhagen, Eben Holderness, and James Pustejovsky. 2023. The Coreference under Transformation Labeling Dataset: Entity Tracking in Procedural Texts Using Event Models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12448–12460, Toronto, Canada. Association for Computational Linguistics.

Evangelia Spiliopoulou, Artidoro Pagnoni, Yonatan Bisk, and Eduard Hovy. 2022. EvEntS ReaLM: Event Reasoning of Entity States via Language Models.

Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. 2020. A Dataset for Tracking Entities in Open Domain Procedural Text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6408–6417, Online. Association for Computational Linguistics.

Jizhi Tang, Yanqiang Qu, and Yansong Li. 2020. Understanding procedural text using interactive entity networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Jingxuan Tu, Timothy Obiso, Bingyang Ye, Kyeongmin Rim, Keer Xu, Liulu Yue, Susan Windisch Brown, Martha Palmer, and James Pustejovsky. 2024. GLAMR: Augmenting AMR with GL-VerbNet Event Structure. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7746–7759, Torino, Italy. ELRA and ICCL.

Zeno Vendler. 1967. Verbs and times. In *Linguistics in Philosophy*, pages 97–121. Cornell University Press.

Veniamin Veselovsky, Manoel Horta Ribeiro, Akhil Arora, Martin Josifoski, Ashton Anderson, and Robert West. 2023. Generating faithful synthetic data with large language models: A case study in computational social science. *arXiv preprint arXiv:2305.15041*.

Xueqing Wu, Sha Li, and Heng Ji. 2023. OpenPI-C: A Better Benchmark and Stronger Baseline for Open-Vocabulary State Tracking. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7213–7222, Toronto, Canada. Association for Computational Linguistics.

Li Zhang, Hainiu Xu, Yue Yang, Shuyan Zhou, Weiqiu You, Manni Arora, and Chris Callison-Burch. 2024. OpenPI2.0: An improved dataset for entity tracking in texts. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 415–431. Association for Computational Linguistics.

A. Silver Annotation Prompt

The following prompt and few-shot examples are used for all ten LLMs in the silver annotation ensemble.

System Prompt

```
You classify entity identity
changes in procedural text.

Given a procedural goal, the
steps completed so far, the cur-
rent step, and a list of entities
with their observed attribute
changes, classify each entity's
identity status at this step.

Labels (use exactly one per en-
tity):

- transform: the entity's TYPE
changes while its identity is
preserved. You would call it by
a different name afterward. Ex-
amples: raw dough becomes baked
bread, whole onion becomes diced
onion, empty bowl becomes bowl
of batter. Moving an entity to
a different location or changing
its temperature alone is NOT a
transform.

- created: the entity comes into
existence at this step; it did
not exist as a distinct object
before.

- destroyed: the entity ceases
to exist at this step; it is used
up, discarded, or absorbed into
something else.

- unchanged: the entity is a
physical object that partici-
pates in the step but is not
meaningfully altered. It is the
same kind of thing before and
after. Tools being used (knife,
oven, bowl), entities that merely
change location, and entities
with superficial attribute dif-
ferences (e.g. same temperature)
are unchanged.

- irrelevant: the entity is not
actually involved in this step,
or is not a physical object. Use
for: agents/users performing
the procedure (e.g. "you", "your
hands"), abstract concepts (e.g.
"knowledge", "confidence"), and
entities whose attribute change
is nonsensical given the step.

For each entity listed, judge its
status at the current step. Re-
spond with exactly one line per
entity in the format:

entity_name: label

Do not add explanation or commen-
tary.
```

Few-shot example (1 of 5)

User:

```
Goal: Make Mango Marinade
Procedure so far: Prepare and
measure all ingredients. / Add
all the ingredients to a food
processor.
Current step: Blend thoroughly
until you achieve a smooth tex-
ture.
Entities:
ingredients: texture=lumpy-
>smooth, texture=intact->blended
food processor: state=on-
>through blending
mixture: texture=chunky-
>smoother
```

Assistant:

```
ingredients: transform
food processor: unchanged
mixture: transform
```

The complete set of five few-shot examples is available in the released codebase.