Event Ontology Expansion via LLM-Based Conceptualization

Anonymous ACL submission

Abstract

Event ontology expansion aims to discover emerging event types from data and extend them to appropriate positions in the existing event ontology.. Existing methods typically cluster contextualized trigger representations and attach induced clusters to the ontology based on instance-level similarity. However, ontology expansion requires concept-level semantics that characterize event types, whereas contextualized trigger representations often conflate these semantics with surface contextual variation, leading to unstable clustering and unreliable hierarchy expansion. To address this issue, we propose ConceptE, a conceptualization-enhanced framework for event ontology expansion. ConceptE first derives concept-level semantics by prompting an LLM with the sentence and event trigger, producing a concise concept name and a natural-language description. It then jointly encodes these semantics with trigger information to build concept-enhanced representations aligned with ontology-level reasoning. This representation design supports more coherent event clustering, more reliable hierarchy expansion, and ontology-consistent type naming. Experiments on ACE, ERE, and MAVEN demonstrate that ConceptE consistently outperforms state-of-the-art approaches across all subtasks of event ontology expansion. In particular, it achieves improvements of up to 12.37% in BCubed-F1 for event clustering and 6.48% in Taxo_F1 for hierarchy expansion, demonstrating the effectiveness of the proposed ConceptE method. The code is available at `https://anonymous.4open.science/r/ConceptE-code-14DF`.

1 Introduction

Event extraction is a core task in natural language processing, with broad applications in information retrieval, knowledge graph construction, and recommendation (Glavaš and Šnajder, 2014; Liu et al., 2017; Wu et al., 2019). Most existing event extraction systems operate under a predefined event ontology, where event types and their hierarchical relations are manually specified in advance. However, such fixed ontologies are difficult to maintain as new data continuously introduces novel event types. This motivates *event ontology expansion*, which aims to automatically discover new event types from data and integrate them into appropriate positions within an existing ontology hierarchy.

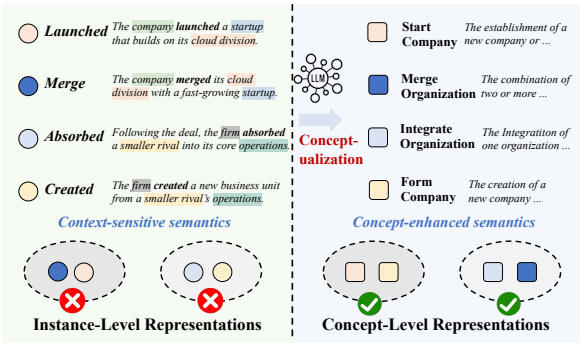


Figure 1: Illustration of the impact of contextual variation on instance-level representations and the stabilizing effect of concept-enhanced representations.

Event ontology expansion can be viewed as a pipeline with three coupled subtasks: event clustering, hierarchy expansion, and type naming. First, unlabeled event mentions are grouped into clusters, where each cluster corresponds to a candidate emerging event type. Second, the discovered event types are attached to appropriate positions in the existing ontology hierarchy. Finally, each newly inserted type is assigned an ontology-consistent name. This formulation follows recent event ontology completion work (Cao et al., 2023) and provides the basis for analyzing the limitations of existing instance-centric methods.

Despite recent progress, existing approaches, including HALTON (Cao et al., 2023), remain largely instance-centric. These methods encode event in-

stances with contextualized trigger representations and use the resulting instance-level similarity for both clustering and hierarchy expansion. As illustrated in Figure 1 (left), this design is vulnerable to surface contextual variation. Triggers appearing in similar organization-centered contexts (e.g., *"launched"* and *"merged"* with company-related words) may be grouped together although they express different event types, while semantically related triggers (e.g., *"absorbed"* and *"created"*) may be separated by lexical or contextual differences.

For clustering, this sensitivity can blur the boundaries between event types. Contextualized trigger representations capture useful trigger cues, but they can overemphasize surrounding entities, lexical cues, and syntactic patterns within the sentence. As a result, clustering may reflect surface similarity more than the concept-level semantics that characterize event types.

For hierarchy expansion, the same reliance on instance-level similarity makes it difficult to identify the correct parent type. Attaching a discovered type requires assessing whether a candidate parent represents a broader event category than the new type, rather than selecting a node by embedding similarity alone. Instance-level similarity may assign high scores to semantically related but structurally incorrect nodes, such as siblings or ancestors. This motivates a hierarchy expansion module that models directed parent-child attachment instead of undirected similarity matching.

To address these limitations, we use LLMs to derive concept-level semantics for event triggers. Given a sentence and its event trigger, an LLM generates a concise concept name and a natural-language description that summarize the event type expressed by the trigger. The derived concept-level semantics are encoded together with the original trigger information to build concept-enhanced representations. As illustrated in Figure 1 (right), these representations group context-diverse trigger occurrences by shared event-type meaning (e.g., *"Start Company"*, *"Merge Organization"*), enabling more consistent clustering and, in turn, more reliable hierarchy expansion.

We propose ConceptE, a conceptualization-enhanced framework for event ontology expansion. ConceptE first derives concept-level semantics with LLMs and encodes them together with the original trigger information. With these concept-enhanced representations, ConceptE clusters event mentions into candidate event types while reducing the effect of surface contextual variation across sentences.

For hierarchy expansion, ConceptE further uses concept-enhanced representations to align newly discovered event types with event types already organized in the existing ontology. Since hierarchy insertion requires identifying a valid parent type rather than merely finding a similar type, ConceptE models parent-child attachment as a directed relation. This design places new event types into structurally appropriate ontology positions by using concept-level semantics and directed attachment, rather than symmetric instance-level similarity alone. The derived concepts also provide semantic signals for ontology-consistent naming.

We evaluate ConceptE on ACE, ERE, and MAVEN, three widely used benchmarks. Experimental results demonstrate that ConceptE consistently outperforms state-of-the-art approaches across subtasks of event ontology expansion. In particular, it achieves improvements of up to 12.37% in BCubed-F1 for event clustering and 6.48% in Taxo_F1 for hierarchy expansion, validating the effectiveness of conceptualization for robust event ontology expansion.

- We propose ConceptE, a conceptualization-enhanced framework that derives concept-level semantics for event triggers and builds concept-enhanced representations for more stable clustering and hierarchy expansion.
- We design a hierarchy expansion module that composes concept-enhanced representations into ontology node representations and learns a directional parent-child linker for directed attachment.
- We conduct extensive experiments on ACE, ERE, and MAVEN, demonstrating that ConceptE consistently outperforms strong baselines in event clustering, hierarchy expansion, and event type naming.

2 Task Definition

Following HALTON (Cao et al., 2023), event ontology expansion takes an existing event ontology $\mathcal{T}_{\text{known}}$, labeled instances of known types $\mathcal{D}_{\text{l}} = \{(x_i, y_i)\}$, and unlabeled instances $\mathcal{D}_{\text{u}} = \{x_j\}$ that may contain unknown event types. The goal is to produce an expanded ontology $\mathcal{T}_{\text{expanded}}$ by discovering novel event types from $\mathcal{D}_{\text{u}}$, attaching

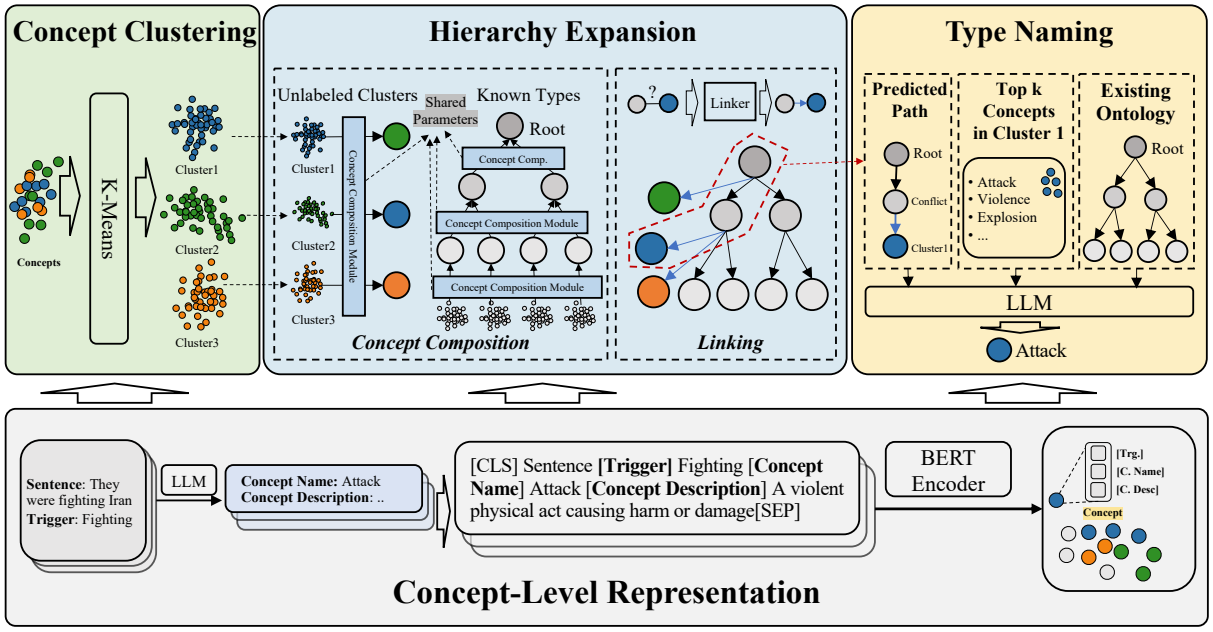


Figure 2: Overall framework of ConceptE.

them to appropriate positions in $\mathcal{T}_{\text{known}}$, and generating ontology-consistent type names. Thus, the task jointly involves event clustering, hierarchy expansion, and type naming.

3 Method

We propose **ConceptE**, a conceptualization-enhanced framework that derives concept-level semantics for event ontology expansion. ConceptE first derives concept-level semantics for event triggers with LLMs, and then applies concept-enhanced representations to three subtasks: (i) *Concept Clustering*, (ii) *Hierarchy Expansion*, and (iii) *Type Naming*. Figure 2 illustrates the overall architecture.

3.1 Concept-Level Representation

Contextualized trigger representations provide important trigger-level information, but they are sensitive to surrounding entities and syntactic patterns within the sentence, which can scatter instances of the same event type and mislead downstream hierarchy expansion. ConceptE enriches these representations by leveraging an LLM to infer the concept-level semantics expressed by each event trigger. Specifically, the prompt provides only the sentence and the event trigger and instructs the LLM to summarize the trigger occurrence into a concise *concept name* and a natural-language *concept description*. The concept name serves as a compact candidate type phrase, while the description explains the event semantics grounded in the sentence containing the trigger. The full prompt and granularity constraints are provided in Appendix D.

Based on the generated concepts, we concatenate the original sentence and trigger information with the generated concept-level semantics:

> [CLS] sentence [Trigger] trigger [Concept Name] short descriptor [Concept Description] semantic abstraction [SEP]

This input retains the original sentence and trigger as direct textual information while representing concept-level semantics through the generated concept name and description. We encode it using a pretrained contextual encoder (e.g., BERT (Devlin et al., 2019)), where the sentence, trigger, concept name, and concept description are jointly modeled through self-attention. Let $\mathbf{h}_{\text{t}}$, $\mathbf{h}_{\text{cn}}$, and $\mathbf{h}_{\text{cd}}$ denote the pooled representations of the trigger span, the generated concept-name span, and the generated concept-description span, respectively. We concatenate these representations and project them back to the original embedding space:

$$\mathbf{h} = \mathbf{W}_{\text{fuse}} \left[\mathbf{h}_{\text{t}} \,\|\, \mathbf{h}_{\text{cn}} \,\|\, \mathbf{h}_{\text{cd}}\right] + \mathbf{b}_{\text{fuse}}, \qquad (1)$$

where $[\cdot\|\cdot]$ denotes vector concatenation, and $\mathbf{W}_{\text{fuse}} \in \mathbb{R}^{d\times 3d}$ is a trainable projection matrix.

The fused representation $\mathbf{h}$ therefore combines trigger-level information with reusable concept-level semantics, providing a more suitable representation for ontology-level reasoning.

3.2 Concept Clustering

To discover coherent candidate event types under contextual variation, we perform event clustering based on concept-enhanced representations. Following the neighborhood contrastive clustering framework in HALTON (Cao et al., 2023), we jointly leverage labeled and unlabeled instances to learn discriminative clustering objectives over these representations.

Let $\mathbf{h}_i$ denote the concept-enhanced representation of instance x_i obtained from the previous module. Given labeled instances $\mathcal{D}_l = \{(x_i^l, y_i^l)\}$, we apply a standard cross-entropy loss:

$$\mathcal{L}_{\text{ce}} = -\frac{1}{|\mathcal{D}_l|} \sum_i y_i^l \log \text{softmax}(\mathbf{h}_i^l). \quad (2)$$

For unlabeled instances $\mathcal{D}_u = \{x_i^u\}$, we update pseudo cluster assignments at the beginning of each epoch. Specifically, we encode all unlabeled instances into the clustering space and run k-means with $k = |\mathcal{Y}_{\text{new}}|$. The resulting pseudo label $\hat{y}_i$ is stored with each unlabeled instance and used to construct pairwise supervision: $q_{ij} = 1$ if $\hat{y}_i = \hat{y}_j$, and $q_{ij} = 0$ otherwise. Following HALTON (Cao et al., 2023), we adopt a hinge-style contrastive objective with pairwise supervision to encourage instances from the same pseudo cluster to produce similar predictions:

$$\mathcal{L}_{\text{hng}} = \sum_{i,j} \Big(q_{ij} d_{ij} + (1 - q_{ij}) \max(0, \alpha - d_{ij}) \Big), \quad (3)$$

where d_{ij} is the bidirectional KL divergence between the predicted cluster distributions of x_i and x_j, and α is the margin for pairs from different pseudo clusters.

To further enhance representation compactness, we introduce neighborhood contrastive losses for both unlabeled and labeled data. For unlabeled instances, positives in Eq. 4 are determined by nearest-neighbor mining rather than gold labels. We maintain a memory bank of current instance representations and retrieve the top-K nearest neighbors for each unlabeled instance. Within a mini-batch, an unlabeled instance x_j is treated as a positive of x_i if the global index of x_j belongs to this top-K neighbor set $\mathcal{N}_i$; the instance itself is also included to form the diagonal positive. In training, we additionally sample one neighbor from $\mathcal{N}_i$ as a second view for supervised contrastive learning. The unlabeled neighborhood contrastive loss is defined as:

$$\mathcal{L}_{\text{ncu}} = -\sum_i \sum_{j \in \mathcal{N}_i} \log \frac{\exp(r_{ij}^u / \tau)}{\sum_{k \neq i} \exp(r_{ik}^u / \tau)}. \quad (4)$$

$$r_{ij}^u = \text{sim}(\mathbf{h}_i^u, \mathbf{h}_j^u). \quad (5)$$

For labeled instances, positives are all in-batch instances sharing the same gold type:

$$\mathcal{L}_{\text{ncl}} = -\sum_i \sum_{j \in \mathcal{N}_i^l} \log \frac{\exp(r_{ij}^l / \tau)}{\sum_{k \neq i} \exp(r_{ik}^l / \tau)}. \quad (6)$$

$$r_{ij}^l = \text{sim}(\mathbf{h}_i^l, \mathbf{h}_j^l). \quad (7)$$

These objectives jointly encourage intra-type compactness and inter-type separability under both supervised and unsupervised settings.

$$\mathcal{L}_{\text{cluster}} = \mathcal{L}_{\text{ce}} + \mathcal{L}_{\text{hng}} + \mathcal{L}_{\text{ncu}} + \mathcal{L}_{\text{ncl}}. \quad (8)$$

3.3 Hierarchy Expansion

After event clustering, each cluster corresponds to a latent event type that must be integrated into the existing ontology. A naive approach attaches clusters based on instance-level similarity. However, such similarity does not explicitly represent semantic scope or attachment direction, and therefore often results in structurally inconsistent expansions. Hierarchy expansion requires deciding whether an existing ontology node can subsume a newly discovered event type. To address this requirement, we explicitly model ontology nodes as semantic units and learn directed parent-child relations based on concept-enhanced representations, allowing newly induced event types to be attached to structurally and semantically appropriate positions.

3.3.1 Concept Composition

To enable hierarchy-aware linking, we introduce a Concept Composition module that constructs semantic representations for ontology nodes in a bottom-up manner. We model the event ontology as a rooted tree, where leaf nodes correspond to known event types or newly discovered clusters and are initialized from their associated instance representations, while internal nodes recursively compose semantics from their children. Newly discovered clusters and known event types are treated uniformly as leaf nodes, and the same Concept Composition module with shared parameters is applied across all hierarchy levels to ensure consistent semantic composition.

Let $\mathcal{C}(v)$ denote the children of node v. We implement composition with attention-weighted pooling over child embeddings and combine it with residual mean pooling:

$$\tilde{\mathbf{h}}_v = \sum_{u \in \mathcal{C}(v)} \alpha_u \mathbf{h}_u, \tag{9}$$

$$\mathbf{h}_v = \mathrm{LayerNorm}\big(\mathrm{Proj}(\tilde{\mathbf{h}}_v) + \mathrm{Mean}(\{\mathbf{h}_u\})\big), \tag{10}$$

where α_u is a learned attention weight normalized over children and $\mathrm{Proj}(\cdot)$ is a linear projection layer. Through this recursive composition process, explicit semantic representations are obtained for all ontology nodes.

3.3.2 Linking

To model the asymmetric semantic relation that a child event type is subsumed by its parent type, we design a directional parent-child linker. Given normalized child and parent embeddings $\mathbf{u}$ and $\mathbf{v}$, we construct interaction features:

$$\mathbf{z} = [\mathbf{u}; \mathbf{v}; \mathbf{u} \odot \mathbf{v}; |\mathbf{u} - \mathbf{v}|], \tag{11}$$

where $[\cdot;\cdot]$ denotes concatenation and $\odot$ denotes elementwise product. A lightweight MLP then produces a directed compatibility score $s(c \to p)$.

For each newly discovered cluster represented by its embedding $\mathbf{h}_{\text{new}}$, we perform greedy top-down inference starting from the ontology root. At each step, we score the new type against each child of the current node as a candidate parent, descend to the best-scoring child if it improves over the current node, and stop otherwise. The final node determines the attachment position of the new event type under greedy top-down inference.

3.3.3 Training Objectives

We train the hierarchy module with two objectives. The composition loss aligns composed node embeddings with gold type-name embeddings, while the linker loss uses existing parent-child edges as positives and non-ancestor nodes as negatives. The overall objective is:

$$\mathcal{L}_{\text{taxo}} = \lambda_{\text{comp}}\, \mathcal{L}_{\text{comp}} + \lambda_{\text{link}}\, \mathcal{L}_{\text{link}}, \tag{12}$$

where $\mathcal{L}_{\text{comp}}$ is a cosine alignment loss and $\mathcal{L}_{\text{link}}$ is a weighted binary cross-entropy loss.

3.4 Type Naming

After clustering and hierarchy expansion, each newly discovered cluster corresponds to a latent event type that requires an ontology-consistent name. The goal of this stage is to convert the latent cluster into a human-readable event type label that matches the granularity and naming style of the existing ontology. This is challenging because a cluster may contain diverse triggers and contexts, and a single representative instance may not fully capture the cluster semantics.

We therefore perform ontology-aware type naming at the cluster level. For each discovered cluster, we aggregate high-frequency concept names as semantic anchors and combine them with the predicted ontology path as structural context. The concept set summarizes what the cluster expresses, while the parent path constrains the naming space to an appropriate ontology region. An LLM then generates a concise type name conditioned on both sources of information. This design encourages names that are semantically faithful to the cluster and structurally consistent with the ontology. Detailed prompt fields and examples are provided in Appendix C.

4 Experimental Setup

4.1 Datasets

We follow the dataset construction in HALTON (Cao et al., 2023) based on ACE (Doddington et al., 2004), ERE (Song et al., 2015), and MAVEN (Wang et al., 2020). For ACE and ERE, the top 10 most frequent types are treated as known, with the remaining types as unknown. For MAVEN, the top 20 are known, and the remaining 40 are unknown, to mitigate long-tail effects. Dataset statistics are provided in Appendix A.

4.2 Baselines

We compare ConceptE with representative baselines for the three subtasks. For event clustering, we include SS-VQ-VAE (Huang and Ji, 2020), ETYPECLUS (Shen et al., 2021), TABS (Li et al., 2021), and HALTON (Cao et al., 2023), covering semi-supervised, unsupervised, abstraction-enhanced, and recent ontology-expansion settings. For hierarchy expansion, we compare with clustering baselines combined with greedy expansion (GE), Type_Similarity, LLMs_Prompt, and HALTON. For type naming, we compare against TABS, T5_Template (Raffel et al., 2020), Trigger_Sel, HALTON, and Top1_Concept. Detailed descriptions are provided in Appendix B.

Table 1: Event clustering performance.

Datasets	Methods	ARI (%)	NMI (%)	Accuracy (%)	BCubed-F1 (%)
ACE	SS-VQ-VAE	8.53	33.81	29.95	27.60
	ETYPECLUS	26.17	53.91	40.70	38.69
	TABS	59.18	79.36	71.42	69.44
	HALTON	67.41	84.29	77.26	75.06
	ConceptE (Ours)	**76.27 (↑ 8.86)**	**89.87 (↑ 5.58)**	**84.58 (↑ 7.32)**	**83.25 (↑ 8.19)**
ERE	SS-VQ-VAE	13.46	40.45	29.96	26.69
	ETYPECLUS	15.89	46.86	34.55	29.13
	TABS	47.22	71.26	60.24	55.82
	HALTON	56.01	78.13	67.72	64.66
	ConceptE (Ours)	**65.61 (↑ 9.60)**	**84.02 (↑ 5.89)**	**75.93 (↑ 8.21)**	**73.33 (↑ 8.67)**
MAVEN	SS-VQ-VAE	3.06	17.57	12.29	11.14
	ETYPECLUS	11.27	30.79	20.82	14.73
	TABS	27.93	53.84	39.38	31.52
	HALTON	36.03	60.34	52.70	39.35
	ConceptE (Ours)	**46.99 (↑ 10.96)**	**69.36 (↑ 9.02)**	**62.81 (↑ 10.11)**	**51.72 (↑ 12.37)**

4.3 Evaluation Metrics

Clustering metrics. Following HALTON (Cao et al., 2023), we evaluate clustering performance using Adjusted Rand Index (ARI) (Hubert and Arabie, 1985), BCubed-F1 (Bagga and Baldwin, 1998), Accuracy (Crouse, 2016), and NMI (Huang and Ji, 2020).

Taxonomy metrics. For hierarchy expansion, we adopt the taxonomy metric (Dellschaft and Staab, 2006) to evaluate parent-child linking accuracy in the ontology, and report Taxo_P, Taxo_R, and Taxo_F1.

Naming metrics. For event type naming, we report ROUGE-L (Lin, 2004) and BERTScore (Zhang* et al., 2020) to measure lexical overlap and semantic similarity between predicted and gold type names.

4.4 Implementation Details

We use BERT-base-uncased (Devlin et al., 2019) as the encoder and Qwen3-32B (Yang et al., 2025) for conceptualization and type naming. Full hyperparameters and training details are provided in Appendix F. We report single-run results due to computational constraints.

5 Results and Analysis

5.1 Event Clustering Results

Table 1 shows that ConceptE consistently outperforms all clustering baselines across ACE, ERE, and MAVEN. Compared with HALTON, ConceptE improves ARI by 8–11% and achieves a 12.37% BCubed-F1 gain on MAVEN, indicating that concept-enhanced representations improve both global cluster alignment and instance-level coherence. These improvements are consistent across datasets with different ontology sizes and unknown-type ratios, suggesting that ConceptE does not rely on dataset-specific type distributions. In particular, the gains on MAVEN are important because MAVEN contains more event types and a more complex ontology, making clustering more vulnerable to contextual variation. These results support the core motivation of ConceptE: modeling concept-level semantics together with trigger information helps reduce within-type variance while preserving discriminative information between types.

5.2 Hierarchy Expansion Results

Table 2 reports hierarchy expansion results under Predicted Cluster and Golden Cluster settings. ConceptE achieves the best Taxo_F1 across datasets and improves over HALTON by 3.35–5.45% with predicted clusters and 2.95–6.48% with golden clusters. The gains under Golden Cluster indicate that ConceptE improves hierarchy modeling beyond the effect of better clustering. Compared with Type_Similarity and LLMs_Prompt, ConceptE benefits from explicitly modeling ontology nodes and directed parent-child relations rather than relying only on embedding similarity or direct prompting. The Predicted Cluster setting reflects the full pipeline, where hierarchy expansion must tolerate upstream clustering errors. The consistent improvements in this setting show that concept-enhanced representations provide more reliable

Table 2: Hierarchy expansion performance.

Datasets	Methods	Predicted Cluster			Golden Cluster		
		Taxo_P (%)	Taxo_R (%)	Taxo_F1 (%)	Taxo_P (%)	Taxo_R (%)	Taxo_F1 (%)
ACE	SS-VQ-VAE+GE	9.12	10.14	9.60	9.52	13.04	11.01
	ETYPECLUS+GE	30.70	23.46	26.59	34.14	33.33	33.73
	TABS+GE	34.31	30.43	32.25	33.33	37.68	35.37
	Type_Similarity	31.79	40.58	35.65	33.33	40.37	36.51
	LLMs_Prompt	34.09	34.78	34.43	42.85	43.47	43.16
	HALTON	37.00	39.13	38.04	44.44	44.92	44.68
	ConceptE (Ours)	**43.48**	**43.48**	**43.48 (↑ 5.44)**	**55.00**	**47.83**	**51.16 (↑ 6.48)**
ERE	SS-VQ-VAE+GE	16.38	14.28	15.26	26.00	25.00	25.49
	ETYPECLUS+GE	9.85	9.52	9.68	18.00	16.66	17.30
	TABS+GE	23.68	17.85	20.36	26.00	25.00	25.49
	Type_Similarity	20.37	21.49	20.88	22.00	21.42	21.71
	LLMs_Prompt	20.68	20.43	20.55	24.00	21.49	22.64
	HALTON	22.54	23.60	23.06	26.80	25.73	26.25
	ConceptE (Ours)	**28.00**	**25.00**	**26.41 (↑ 3.35)**	**32.00**	**28.57**	**30.19 (↑ 3.94)**
MAVEN	SS-VQ-VAE+GE	19.45	20.14	19.79	26.94	43.00	33.13
	ETYPECLUS+GE	15.83	17.50	16.62	23.75	28.75	26.01
	TABS+GE	27.82	32.03	29.78	27.53	40.42	32.75
	Type_Similarity	22.50	27.50	24.75	27.91	32.50	30.03
	LLMs_Prompt	12.50	10.00	11.11	27.50	21.50	23.97
	HALTON	34.79	52.50	41.85	39.38	**59.38**	47.35
	ConceptE (Ours)	**41.25**	**55.42**	**47.30 (↑ 5.45)**	**43.75**	59.17	**50.30 (↑ 2.95)**

cluster semantics for attachment. Meanwhile, the Golden Cluster setting isolates the hierarchy module, and the remaining gains demonstrate that recursive concept composition and directional linking are useful even when clustering is assumed to be correct.

5.3 Naming Results

Table 3 compares ConceptE with naming baselines using ROUGE-L and BERTScore. ConceptE consistently outperforms Top1_Concept and HALTON, showing that multiple representative concepts and predicted ontology paths provide stronger semantic coverage and structural constraints for naming. Top1_Concept performs reasonably well because the most frequent concept often captures the dominant semantics of a cluster. However, relying on a single concept can miss sub-patterns within the cluster or produce names that are too narrow. ConceptE instead conditions generation on multiple representative concepts and the predicted path, which encourages names that match both the cluster semantics and the naming style of the existing ontology.

Table 3: Event type naming performance.

Datasets	Methods	ROUGE-L (%)	BERTScore (%)
ACE	TABS	17.49	29.40
	T5_Template	18.66	35.25
	Trigger_Sel	20.86	42.46
	HALTON	24.09	46.24
	Top1_Concept	26.21	48.72
	ConceptE (Ours)	**30.43(↑ 4.22)**	**53.33(↑ 4.61)**
ERE	TABS	11.90	28.03
	T5_Template	13.46	32.51
	Trigger_Sel	12.59	35.07
	HALTON	16.20	39.32
	Top1_Concept	16.43	39.87
	ConceptE (Ours)	**17.66(↑ 1.23)**	**41.21(↑ 1.34)**
MAVEN	TABS	16.02	36.24
	T5_Template	24.94	38.20
	Trigger_Sel	27.30	40.70
	HALTON	30.89	41.14
	Top1_Concept	31.12	41.53
	ConceptE (Ours)	**33.36(↑ 2.24)**	**43.96(↑ 2.43)**

5.4 Ablation Studies

We conduct ablation studies to assess conceptualization, concept composition, and the parent-child linker. As shown in Table 4, removing concept information consistently degrades clustering performance on ACE, with BCubed-F1 dropping from 83.25% to 74.23%. Table 5 further shows that conceptualization also benefits hierarchy expansion.

Table 4: Ablation study by removing Conceptualization in event clustering on the ACE dataset.

Method	ARI (%)	NMI (%)	Accuracy (%)	BCubed-F1 (%)
ConceptE	**76.27**	**89.87**	**84.58**	**83.25**
w/o Concept	66.20	84.08	76.79	74.23

Table 5: Ablation study by removing Conceptualization, Concept Composition, and Linker in Hierarchy Expansion on the ACE dataset.

Methods	Taxo_F1 (%)	
	Predicted Cluster	Golden Cluster
ConceptE	**43.48**	**51.16**
w/o Concept	34.78	41.86
w/o Comp.	39.13	46.51
w/o Linker	14.35	19.09
w/o Comp.&Linker	36.67	43.01

These drops confirm that concepts are not merely auxiliary text for naming, but also improve the learned representation space used by clustering and hierarchy expansion. Without conceptualization, the model loses explicit concept-level semantics and becomes more susceptible to surface lexical and syntactic variation.

Removing either concept composition or the directional linker also hurts Taxo_F1, with the largest drop from removing the linker. This confirms that recursive node composition and asymmetric parent-child modeling are both important for reliable hierarchy expansion. The sharp degradation without the linker suggests that symmetric similarity is insufficient for hierarchy insertion, where the direction from child to parent matters. The composition module also contributes because internal ontology nodes need explicit semantic representations rather than simple averages over leaf instances.

5.5 Analysis of Interaction Features in the Linker

We further analyze the interaction features used by the parent-child linker in Appendix E. The results show that combining node identity features with compatibility and mismatch features yields the best hierarchy expansion performance. This supports the design of the linker, where parent-child attachment requires more than symmetric embedding similarity.

5.6 Event Clustering Visualization

We further visualize event representations on the ERE dataset with t-SNE under multiple perplexity values. As shown in Appendix G, ConceptE consistently forms more compact and better separated clusters than HALTON, supporting the quantitative improvements in Table 1. Although t-SNE provides only qualitative evidence, the trend remains stable across different perplexity values, suggesting that the benefit of conceptualization is not an artifact of a single visualization setting.

6 Related Work

Early work relevant to event ontology expansion primarily focused on event type induction and ontology induction, which discover latent event types from unlabeled or weakly labeled corpora (Huang and Ji, 2020; Shen et al., 2021; Edwards and Ji, 2023; Xu et al., 2024). These methods reduce reliance on fixed ontologies, but mainly focus on clustering or constructing ontologies from scratch rather than expanding a given ontology with newly discovered event types. Taxonomy expansion methods instead insert provided new concepts into an existing taxonomy (Shen et al., 2020; Yu et al., 2020; Manzoor et al., 2020; Zeng et al., 2025). This assumption is less suitable for event ontologies, where new types often emerge from data.

The most related work, HALTON (Cao et al., 2023), discovers new event types, attaches them to an existing event ontology, and generates type names. However, HALTON remains instance-centric: both clustering and hierarchy expansion rely on contextualized instance representations, which are sensitive to context variation. ConceptE differs by deriving concept-level semantics with LLMs and building concept-enhanced representations for event ontology expansion.

7 Conclusions

This work proposes ConceptE, a conceptualization-enhanced framework for event ontology expansion. ConceptE uses LLMs to derive concept-level semantics for event triggers and jointly encodes them with trigger information to build concept-enhanced representations. By reducing surface contextual variation, ConceptE supports more coherent event clustering and provides reliable semantic units for hierarchy expansion. For hierarchy expansion, ConceptE further uses recursive concept composition and a directional parent-child linker to attach newly discovered event types to structurally appropriate ontology positions. Experiments and ablation studies on three benchmarks demonstrate the effectiveness of conceptualization as a semantic bridge between event triggers and ontology structures.

Limitations

Despite its effectiveness, ConceptE relies on LLMs for event conceptualization and type naming, which may introduce extra preprocessing cost when scaling to very large corpora. The quality of generated concepts can also be affected by model bias or occasional generation inconsistency, although our representation learning module mitigates this risk by jointly encoding the original trigger information and the generated concept-level semantics. Following prior event ontology expansion methods such as HALTON (Cao et al., 2023), our experiments assume that event triggers are given. This setting enables a controlled comparison with existing methods, but future work can integrate automatic trigger detection to support a more end-to-end expansion pipeline.

Ethical Considerations

This work focuses on automatic event ontology expansion from text and does not involve human subjects or personally identifiable information. All experiments are conducted on publicly available benchmark datasets widely used in prior research. The proposed framework organizes event types at an abstract semantic level and does not generate new factual claims about real-world entities.

References

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Pengfei Cao, Yupu Hao, Yubo Chen, Kang Liu, Jiexin Xu, Huaijun Li, Xiaojian Jiang, and Jun Zhao. 2023. Event ontology completion with hierarchical structure evolution networks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 306–320, Singapore. Association for Computational Linguistics.

David F Crouse. 2016. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696.

Klaas Dellschaft and Steffen Staab. 2006. On how to perform a gold standard based evaluation of ontology learning. In *Proceedings of the 5th International Conference on The Semantic Web*, ISWC'06, page 228–241, Berlin, Heidelberg. Springer-Verlag.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Carl Edwards and Heng Ji. 2023. Semi-supervised new event type induction and description via contrastive loss-enforced batch attention. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3805–3827, Dubrovnik, Croatia. Association for Computational Linguistics.

Goran Glavaš and Jan Šnajder. 2014. Event graphs for information retrieval and multi-document summarization. *Expert Syst. Appl.*, 41(15):6904–6916.

Lifu Huang and Heng Ji. 2020. Semi-supervised new event type induction and event detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 718–724, Online. Association for Computational Linguistics.

Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification*, 2(1):193–218.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chun-Yi Liu, Chuan Zhou, Jia Wu, Hongtao Xie, Yue Hu, and Li Guo. 2017. Cpmf: A collective pairwise matrix factorization model for upcoming event recommendation. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 1532–1539.

Emaad Manzoor, Rui Li, Dhananjay Shrouty, and Jure Leskovec. 2020. Expanding taxonomies with implicit edge semantics. In *Proceedings of The Web Conference 2020*, WWW '20, page 2044–2054, New York, NY, USA. Association for Computing Machinery.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi Wang, Kuansan Wang, and Jiawei Han. 2020. Taxoexpan: Self-supervised taxonomy expansion with position-enhanced graph neural network. In *Proceedings of The Web Conference 2020*, WWW '20, page 486–497, New York, NY, USA. Association for Computing Machinery.

Jiaming Shen, Yunyi Zhang, Heng Ji, and Jiawei Han. 2021. Corpus-based open-domain event type induction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5427–5440, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ERE: Annotation of entities, relations, and events. In *Proceedings of the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98, Denver, Colorado. Association for Computational Linguistics.

Laurens van der Maaten. 2014. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15(93):3221–3245.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671, Online. Association for Computational Linguistics.

Xindong Wu, Jia Wu, Xiaoyi Fu, Jiachen Li, Peng Zhou, and Xu Jiang. 2019. Automatic knowledge graph construction: A report on the 2019 icdm/icbk contest. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1540–1545.

Nan Xu, Hongming Zhang, and Jianshu Chen. 2024. CEO: Corpus-based open-domain event ontology induction. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 946–964, St. Julian's, Malta. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*. Presents Qwen3 series (dense MoE, 0.6–235B params), hybrid thinking mode thinking budget mechanism :contentReference[oaicite:4]index=4.

Yue Yu, Yinghao Li, Jiaming Shen, Hao Feng, Jimeng Sun, and Chao Zhang. 2020. Steam: Self-supervised taxonomy expansion with mini-paths. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1026–1035, New York, NY, USA. Association for Computing Machinery.

Qingkai Zeng, Yuyang Bai, Zhaoxuan Tan, Zhenyu Wu, Shangbin Feng, and Meng Jiang. 2025. CodeTaxo: Enhancing taxonomy expansion with limited examples via code language prompts. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4131–4144, Vienna, Austria. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Dataset Statistics

Table 6 reports the split of known and unknown event types used in our experiments. Following HALTON (Cao et al., 2023), we simulate the realistic ontology expansion scenario where only frequent event types are covered by the initial ontology, while less frequent or newly emerging types must be discovered from unlabeled data. This setting evaluates whether a model can expand an existing ontology beyond its predefined type inventory rather than merely classify instances into known categories.

Table 6: Dataset statistics for event ontology expansion.

Dataset	#Known	#Unknown	#Total	Unk. Ratio (%)
ACE	10	23	33	69.7
ERE	10	28	38	73.7
MAVEN	20	40	60	66.7

The unknown-type ratios are high across all three datasets, ranging from 66.7% to 73.7%. Therefore, the evaluation places substantial emphasis on discovering and organizing unseen event types. MAVEN contains the largest number of total and unknown types, making it a more challenging benchmark for both clustering and hierarchy expansion because the model must distinguish a larger set of fine-grained event semantics.

B Baseline Details

We include baselines that cover the three major abilities required by event ontology expansion: discovering unknown types, attaching discovered types to an ontology, and producing human-readable type names. The goal is not only to compare against methods with similar architectures, but also to test

whether conceptualization brings consistent benefits over representative alternatives at each stage.

Event clustering baselines. SS-VQ-VAE (Huang and Ji, 2020) is a semi-supervised event type induction method based on VQ-VAE. ETYPECLUS (Shen et al., 2021) clusters events with salient predicate-argument features in a latent spherical space. TABS (Li et al., 2021) injects abstraction-based semantic representations. HALTON (Cao et al., 2023) adopts neighborhood contrastive learning with K-Means pseudo labels. These baselines allow us to compare ConceptE with semi-supervised clustering, unsupervised clustering, abstraction-based representations, and the strongest directly related event ontology expansion method.

Hierarchy expansion baselines. Clustering baselines + GE combine clustering-based discovery with greedy expansion. Type_Similarity links averaged cluster representations to encoded known type names. LLMs_Prompt directly prompts an LLM to select a parent type. HALTON performs hierarchy-aware representation learning with path-based supervision and greedy top-down inference. This group tests whether explicit hierarchy modeling is needed beyond flat clustering, and whether learned parent-child linking is more reliable than direct embedding similarity or direct LLM prompting.

Naming baselines. TABS uses abstraction-based generation; T5_Template (Raffel et al., 2020) fills predefined templates; Trigger_Sel selects a trigger word from the cluster; HALTON prompts an LLM with a cluster-center instance and predicted path; Top1_Concept selects the most frequent concept name within a cluster. The naming baselines represent template-based generation, trigger-based selection, LLM prompting with instance-level information, and a simple concept-frequency heuristic. Comparing against Top1_Concept is particularly useful because it isolates whether using only the most frequent generated concept is sufficient. The results show that conditioning on multiple representative concepts and the predicted ontology path is more effective.

C Type Naming Details

The type naming stage is formulated as ontology-aware generation. Given a discovered cluster, we provide the LLM with representative concepts from the cluster and its predicted parent path. The representative concepts summarize the cluster semantics, while the parent path exposes the naming convention and semantic scope expected by the ontology. The prompt instructs the LLM to generate a concise event type name rather than a free-form description. This design is important because event type names are not arbitrary summaries: they must match the granularity of the existing ontology and remain compatible with neighboring nodes in the hierarchy. For example, a name that is too broad may overlap with its parent type, while a name that is too specific may describe only a subset of instances in the cluster.

Figure 4 shows the full prompt used for type naming. For completeness, Figure 3 also shows the conceptualization prompt that produces the concept names used by the naming module. The prompt examples are moved to the appendix to keep the main text focused on the modeling idea. They also provide reproducibility details for how concepts and names are elicited from the LLM.

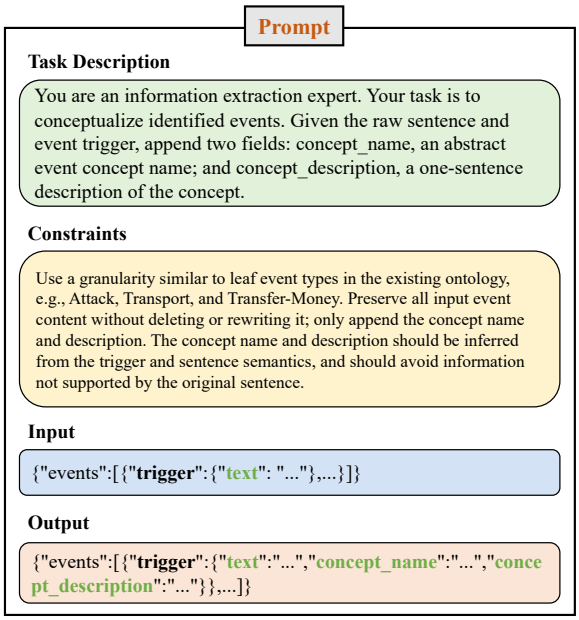


Figure 3: Illustration of the conceptualization prompt. The prompt takes the input sentence and the identified trigger, and instructs the LLM to generate concept names and descriptions for the trigger.

D Conceptualization Details

To reduce generation noise and hallucination, the conceptualization prompt imposes three constraints. First, concept names are required to be concise schema-level categories with granularity compara-

Prompt

Task Description

Given ***Known Ontology***, ***Representative Concepts*** *from Cluster* and ***Parent Path***, please assign event type names with high semantic precision and consistency with the existing ontology.

Existing Ontology Input

Analyze the following event types in existing ontology to learn their naming patterns, granularity, and hierarchical structure:

Root:Life:Die, Root:Contact:Meet, Root:Conflict:Attack, ...

In-Context Examples

Representative Concepts: [Election, Election-Event, Select, ...]
Parent Path: Root: Personnel
Output: Root: Personnel: Elect
...

Incomplete Entry

Representative Concepts: [Marriage, Ceremony, Marital-Union, Marry, Marriage-End]
Parent Path: Root: Life
Output:

Figure 4: An example prompt for ontology-aware event type naming.

ble to event ontology leaf types. Second, the input sentence and trigger must be preserved exactly, and the model should only output the concept name and concept description. Third, the concept name and description must be grounded in the sentence containing the trigger rather than unsupported external information.

Concept granularity is important for ontology expansion. If concepts are too coarse, different unknown event types become hard to separate; if they are too fine-grained, the representation again becomes sensitive to entities, time expressions, or incidental details. We therefore use concepts at the level of candidate event-type phrases: more abstract than a surface trigger, but not tied to a specific instance.

ConceptE uses conceptualization as an intermediate semantic abstraction rather than as a replacement for the original event trigger. The original sentence and trigger preserve trigger-level information, while the generated concept exposes the concept-level semantics that are often implicit in the sentence. Jointly encoding these signals helps the model retain discriminative lexical cues while reducing sensitivity to incidental contextual variation.

In practice, the generated concept description is useful when the concept name alone is ambiguous. For example, a short concept such as *Transfer* may correspond to different event semantics depending on whether the context involves people, ownership, money, or organizational control. The description provides a grounded semantic explanation, allowing the encoder to distinguish such cases more reliably.

E Feature Analysis

The parent-child linker predicts whether a newly discovered type should be attached under a candidate ontology node. Since this relation is directional, a pure similarity score is insufficient. A child type and a parent type may be semantically related, but the direction of subsumption still needs to be modeled. We therefore compare several interaction feature combinations in Table 7.

Table 7: Comparison of different interaction feature combinations in the parent-child linker on the ACE dataset.

Methods	Taxo_F1 (%)	
	Predicted Cluster	Golden Cluster
$[\mathbf{u}; \mathbf{v}; \mathbf{u} \odot \mathbf{v}; \lvert\mathbf{u} - \mathbf{v}\rvert]$	**43.48**	**51.16**
$[\mathbf{u}; \mathbf{v}; \mathbf{u} \odot \mathbf{v}]$	41.62	48.32
$[\mathbf{u}; \mathbf{v}; \lvert\mathbf{u} - \mathbf{v}\rvert]$	38.36	44.82
$[\mathbf{u}; \mathbf{v}]$	35.56	40.21

The full feature set performs best in both predicted-cluster and golden-cluster settings. Concatenation preserves the identity of the child and parent representations, element-wise product captures compatibility, and absolute difference captures mismatch between the two nodes. Removing either product or difference features degrades performance, indicating that parent-child linking benefits from both similarity-oriented and distance-oriented interactions. Using only concatenation performs worst, which suggests that the linker needs explicit cross-node interaction features rather than relying solely on an MLP to infer all pairwise relations from raw embeddings.

F Implementation Details

This section provides additional training and inference details for reproducibility. All modules are trained under the same dataset splits as HALTON, and the number of unknown clusters is set to the number of unknown event types in each benchmark. Unless otherwise stated, hyperparameters are selected on the development set and then kept fixed for the reported test results.

Event Clustering. For event clustering, we use BERT-base-uncased (Devlin et al., 2019) as the

encoder and use the 12th-layer hidden states for representation extraction. The maximum input length is 240, the hidden dimension is 512, and the clustering space dimension is 256. Trigger spans and concept spans are pooled with max pooling. We train the clustering module with Adam using a learning rate of 1×10^{-4}, batch size 64, and 60 epochs. The supervised contrastive temperature is 0.07, and the center loss weight is 0.005. For unlabeled neighborhood mining, $K = 2$ for ACE and ERE and $K = 15$ for MAVEN. The k-means pseudo labels are refreshed once per epoch with the number of clusters set to the number of unknown event types. Event conceptualization is performed using Qwen3-32B (Yang et al., 2025). During concept generation, we set temperature to 0.7, top_p to 0.8, top_k to 20, the maximum token budget to 8192, and disable the model's thinking mode. The memory bank for neighborhood mining is updated with the latest instance representations at each epoch. This allows nearest-neighbor positives for unlabeled data to reflect the current representation space rather than a fixed initialization. We use a larger neighborhood size for MAVEN because it contains more unknown event types and more diverse event mentions.

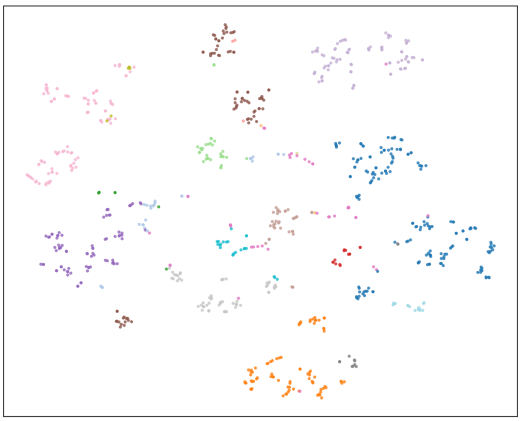
Hierarchy Expansion. For hierarchy expansion, we reuse the BERT encoder trained in the clustering stage and freeze all its parameters. We then train only the hierarchy modules, including the recursive concept composition module and the directional parent-child linker. The hierarchy stage uses Adam with a learning rate of 1×10^{-3}, batch size 256, and 100 epochs. The cosine alignment loss for the composition module is weighted by 0.1. For hierarchy prediction, we use weighted binary cross-entropy with positive and negative edge weights both set to 2.0. For each child node, up to 10 negative parent nodes are sampled from non-ancestor nodes; when available, same-level non-ancestor nodes are preferred as hard negatives. We use the trained clustering checkpoint to obtain cluster representations for the predicted-cluster setting and gold labels for the golden-cluster setting. The predicted-cluster setting evaluates the full pipeline, where hierarchy expansion must handle errors propagated from event clustering. The golden-cluster setting isolates the hierarchy module by removing clustering errors, allowing us to test whether the ontology modeling component itself is effective.

Type Naming. For type naming, we employ Qwen3-32B with the same decoding configuration as in conceptualization. The naming prompt takes the existing ontology context, high-frequency concept names from the target cluster, and the predicted parent path as input. We use high-frequency concepts because they are more likely to represent the shared semantics of the cluster rather than noisy outlier mentions. The predicted parent path is included to encourage names that are consistent with the semantic scope and naming style of the target ontology region.

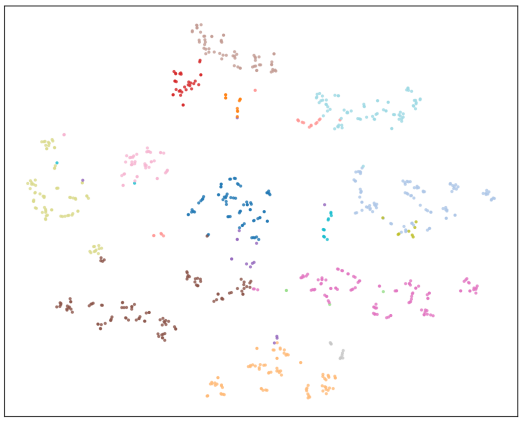
G Event Clustering Visualization

To provide an intuitive comparison of clustering quality and examine whether the visualization trend is stable across t-SNE hyperparameters, we project event representations on the ERE dataset into two dimensions using t-SNE (van der Maaten, 2014). Because t-SNE is sensitive to perplexity, we set the perplexity to 10, 30, and 50, and visualize the event representations of HALTON and ConceptE under each setting. Figure 5 shows six results: (a), (c), and (e) correspond to HALTON under the three perplexity values, while (b), (d), and (f) correspond to ConceptE under the same settings. Across different perplexity values, HALTON shows less compact clusters and less clear boundaries, whereas ConceptE produces more stable local groups and clearer separation. This qualitative trend supports the quantitative clustering results in Table 1.

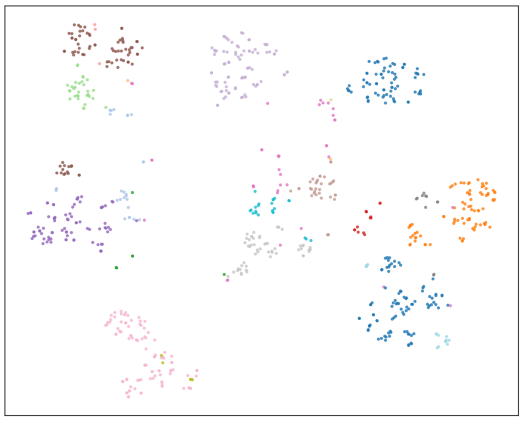
We emphasize that t-SNE visualizations should be interpreted as qualitative evidence only. Distances and cluster shapes in two-dimensional projections can be affected by random initialization, perplexity, and optimization dynamics. For this reason, our main conclusions rely on quantitative clustering metrics, while the visualization is used to illustrate that the improvement brought by conceptualization is also visible under multiple projection settings.



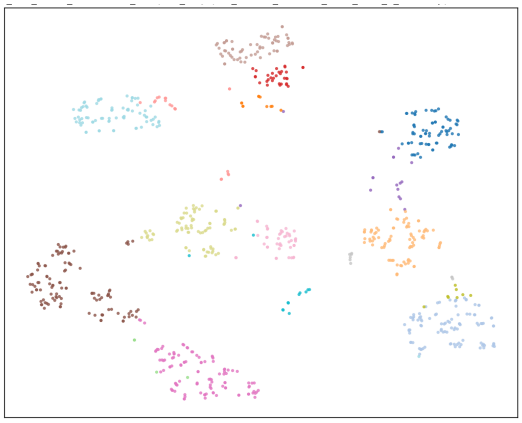
(a) HALTON (Perplexity=10)



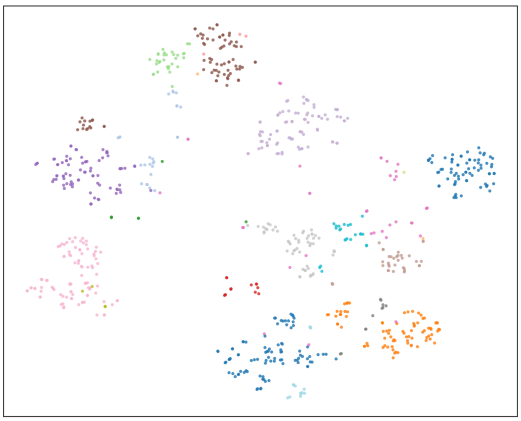
(b) ConceptE (Perplexity=10)



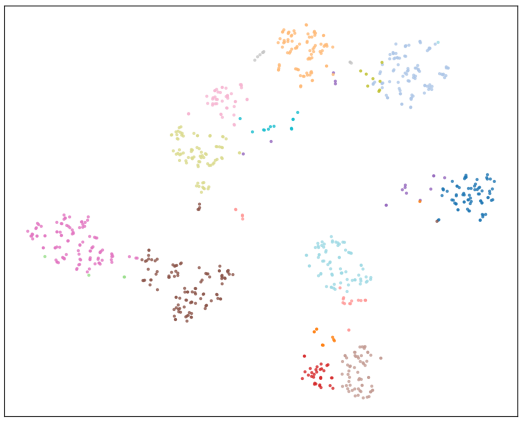
(c) HALTON (Perplexity=30)



(d) ConceptE (Perplexity=30)



(e) HALTON (Perplexity=50)



(f) ConceptE (Perplexity=50)

Figure 5: Visualization of event clustering features under different t-SNE perplexity values.