Semantically-informed Hierarchical Event Modeling

Shubhashis Roy Dipta, Mehdi Rezaee, Francis Ferraro
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore, MD 21250 USA
{sroydip1,rezaee1,ferraro}@umbc.edu

Abstract

Prior work has shown that coupling sequential latent variable models with semantic ontological knowledge can improve the representational capabilities of event modeling approaches. In this work, we present a novel, doubly hierarchical, semi-supervised event modeling framework that provides structural hierarchy while also accounting for ontological hierarchy. Our approach consists of multiple layers of structured latent variables, where each successive layer compresses and abstracts the previous layers. We guide this compression through the injection of structured ontological knowledge that is defined at the type level of events: importantly, our model allows for partial injection of semantic knowledge and it does not depend on observing instances at any particular level of the semantic ontology. Across two different datasets and four different evaluation metrics, we demonstrate that our approach is able to out-perform the previous state-of-the-art approaches by up to 8.5%, demonstrating the benefits of structured and semantic hierarchical knowledge for event modeling.

1 Introduction

Intuitively, there is a hierarchical nature to complex events: e.g., on Fig. 1, there are two events, one involves going to the hospital and another one is getting treatment. Even if important portions may differ, but these two situations have one abstract concept in common: **Cure** (of a disease). Clearly, there is a connection among the events reported in a situation and they all contribute to a bigger goal ("Cure" in this case). The main purpose of our work is to exploit this nature of connection to improve event modeling. However, much like linguistic structure, this event structure is generally not directly observed, making it difficult to learn event models that reflect this hierarchical nature.

For high-level inspiration, we look to past approaches in syntactic modeling (Collins, 1997; Klein and Manning, 2003; Petrov et al., 2006): we can approach hierarchical event modeling through structured learning, or through richer (semantic) data. A structural approach accounts for the hierarchy as part of the model itself, such as with hierarchical random variables (Cheung et al., 2013; Ferraro and Van Durme, 2016; Weber et al., 2018; Huang and Ji, 2020; Gao et al., 2022). Richer semantic data provides hierarchical knowledge, such as event inheritance or composition, as part of the data made available to the model and learning algorithm (Botschen et al., 2017; Edwards and Ji, 2022; Zhang et al., 2020).

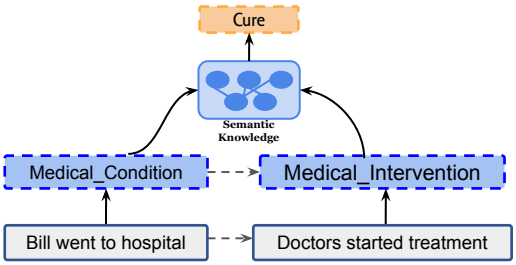

Figure 1: Complex events can be hierarchical. The purple boxes represent the events themselves (as would be reported in a news story). Blue dashed boxes are annotated semantic frames & the orange dashed box is the more abstract, general frame connecting the "Medical_Condition" and "Medical_Intervention" events. Events and frames are sequentially connected.

In this work, we provide an approach that addresses both of these notions of hierarchical event modeling jointly. Fundamentally, our model is an encoder-decoder based hierarchical model comprised of two layers of semi-supervised latent variable sequence. The first layer encodes the events to semantic frames and the next layer compresses down the semantic frames to a more abstract concept. We call these the base and compression layers, respectively. The base layer operates over the event sequence (the gray boxes in Fig. 1); when available, our base layer also considers auxiliary semantic in-

*Proceedings of the The 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 353–369
July 13-14, 2023 ©2023 Association for Computational Linguistics

formation, such as automatically extracted semantic frames (the blue dashed boxes in Fig. 1). Meanwhile, the compression layer compresses down the semantic frames to a more abstract concept (orange dashed box in Fig. 1) using an existing structued semantic resource (in our paper, FrameNet). Our work can be thought of as extending previous work in semi-supervised event modeling (Rezaee and Ferraro, 2021) to account for both structural and semantic hierarchy.

Joining both the structural and semantic approaches together poses a number of challenges. First, getting reliable, wide-coverage semantic event annotations can be a challenge. Development of semantic annotation resources is time consuming and expensive (Baker et al., 1998; O'Gorman et al., 2016).[1] Part of our solution should leverage existing semantic annotation resources.

Second, although event extraction capabilities have steadily improved, enabling automatically produced annotations to be used directly (Padia et al., 2018; Huang and Huang, 2021), these tools still produce error-laden annotation, especially on out-of-domain text. While rich latent variable methods have been previously developed, adapting them to make use of noisy event extractions can be a challenge. Our learning approach must still be able to handle imperfect extractions. Recent work has shown how neural sequence approaches can do so (Rezaee and Ferraro, 2021), but there remains a question of how to generalize this. Part of our solution should allow for hierarchical semi-supervision.

We present a hierarchical latent variable encoder-decoder approach to address these challenges. We ground our work in the FrameNet semantic frame ontology (Baker et al., 1998), from which we extract possible abstract frames from sequences of inferred (latent) frames. This lets us leverage existing semantic resources. We develop a semi-supervised, hierarchical method capable of handling noisy event extractions. Our approach enables learning how to represent more abstract frame representations. Our contributions are:

- We provide a novel, hierarchical, semi-supervised event learning model.
- We show how to use an existing rich semantic frame resource (FrameNet) to provide both observable event frames and less observable abstract frames in a neural latent variable model.
- Our model can use FrameNet to give a more informed signal by leveraging compression of events when predicting what event comes next, what sequence of events follows an initial event, and missing/unreported events.
- With pre-training only, our model can generate event embeddings that better reflect semantic relatedness than previous works, evincing a zero-shot capability.
- We perform comprehensive ablations to show the importance of different factors of our model.

Our code is available at https://github.com/dipta007/SHEM.

2 Related Works

Our work draws on event modeling, latent generative modeling, lexical and semantic knowledge ontologies, and hierarchical modeling.

2.1 Event Modeling

There have been several efforts to understand events and their relationships with broader semantic notions. Previous research has explored the use of hierarchical models based on autoencoders for script generation, such as the work of Weber et al. (2018). In contrast to their work, instead of a chain-like hierarchy, we have used a multi-layer hierarchy to compress the events to abstract processes. Additionally, our approach allows for semi-supervised training, if such labels are available. Our work has shown that using semi-supervision helps the model to generalize better on both layers. In a related study, Rezaee and Ferraro (2021) used the Gumbel-Softmax technique and partially observed frames to model event sequences and generate contextualized event frames. While their approach is capable of generalizing each event in a sequence, the number of predicted frames in the sequence is equivalent to the number of events. Thus, unlike our approach, it was not designed to compress or generalize the overall event sequence.

Bisk et al. (2019) demonstrated the effectiveness of event modeling for generating a concrete concept from an abstract one, using the example of cooking. Several studies in recent years have utilized event modeling to predict event types (Chen et al., 2020; Pepe et al., 2022; Huang and Ji, 2020). These studies focus on identifying the action and

[1] While prompt-based label semantics (Hsu et al., 2022; Huang et al., 2022) are recent successful ways of enabling lower resource learning, these generally are tied to specific tasks and may be limited by what exemplars are given.

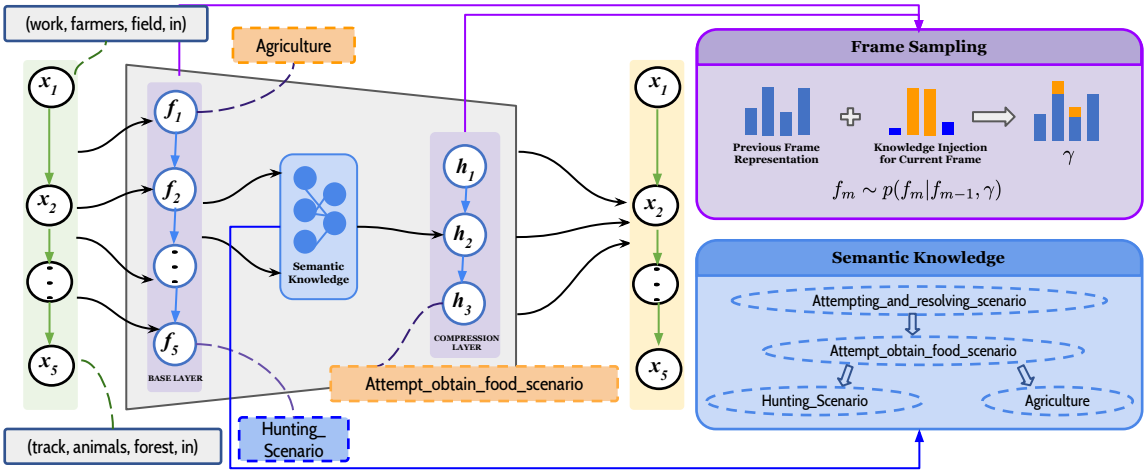

Figure 2: An overview of **S**emantically-informed **H**ierarchical **E**vent **M**odeling (SHEM). The orange dashed boxes are observed frames & the blue dashed boxed are masked frames. Top right: a frame is sampled with the injection of observed frames. Bottom right: semantic knowledge graph is shown for 4 nodes with only "Inheritance" relations.

object involved in an event, where the action represents the activity being performed and the object is the entity affected by the action.

2.2 Latent Generative Modeling

Latent generative modeling is a widely-used method for representing data x through the use of high-level, hidden representations f. Specifically, we express the joint probability $p(x, f)$ as $p(x, f) = p(x|f)p(f)$. Especially when f is not fully observed, this factorization can productively be thought as a soft grouping or clustering of the data in x. This equation will serve as the foundation for our approach.

Maximizing log-likelihood is known to be computationally challenging in this context. Kingma et al. (2014) later used a variational autoencoder (Kingma and Welling, 2013, VAE) in a semi-supervised manner to learn latent variables, dividing the dataset into observed and unobserved labels. In our case, instances are partially observed (rather than fully observed or not). Huang and Ji (2020) used a VAE both to prevent overfitting on seen event types and to enable prediction of novel types.

2.3 Lexical and Semantic Resources

Multiple resources, such as PropBank (Gildea and Jurafsky, 2002), OntoNotes (Hovy et al., 2006), AMR (Banarescu et al., 2013), VerbNet (Schuler, 2005), and FrameNet (Baker et al., 1998), provide annotations related to event semantics. Many consider predicate-argument semantics, such as defining who is performing (or experiencing) an event, and various ways that event may occur. FrameNet provides detailed predicate-argument characterizations and multi-faceted relations linking different frames together, such as frame subtyping (e.g., *inheritance*), temporal/causal (e.g., *precedes*, *causative*), and compositionality (e.g., *uses*, *subframe*). Consider the AGRICULTURE frame from Fig. 2: FrameNet defines an *inheritance* relation between it and a ATTEMPT_OBTAIN_FOOD_SCENARIO, which can be thought of as a container grouping together frames all related to a broader scenario of attempting to obtain food, such as HUNTING_SCENARIO. A scenario container frame provides a notion of compositionality, defining potential correlations or alternatives among frames. Due to these rich semantics, we focus on FrameNet in this paper as an exemplar.

Prior research has shown the utility of FrameNet in predicting the relationship between predicates (Aharon et al., 2010; Ferraro et al., 2017); frame-directed claim verification (Padia et al., 2018); and text summarization (Guan et al., 2021; Han et al., 2016; Chowanda et al., 2017). Unfortunately, while document-level frames have been of long-standing interest within targeted domains (Sundheim, 1992, 1996; Ebner et al., 2020; Du et al., 2021), development of task agnostic document-level frames has been limited. E.g., while FrameNet defines these compositional-like scenario frames, annotation coverage is limited: In the FrameNet 1.7 data used to train frame parsers, out of nearly 29,000 fulltext annotations, there are

only 28 **annotated** "scenario" frames.

3 Method

Our core aim is to provide a hierarchical event model that incorporates both structural and semantic hierarchy. We call our model SHEM (**S**emantically-informed **H**ierarchical **E**vent **M**odeling). An overview is in Fig. 2, where an observed event sequence (green x_i) is latently modeled as multiple sequences of semantic frames (f_i and h_j), augmented by a semantic resource.

We examine the strengths and limitations of structural and semantic hierarchy. Our experiments explore the effect of compressing the number of frames on ability to predict what happens next in an event sequence, and, given an initial seed event, how an event sequence is likely to unfold. We also extend our work to show how our model can produce better intrinsic event representations.

3.1 Model Setup

Our model is a sequence-to-sequence hierarchical model (§3.3). It is comprised of two layers (a base and a compression layer) of an encoder & decoder (§3.2). During training (§3.4), we provide the model partially observed semantic frames in the base layer in order to guide it in encoding event sequences into latent variables. In the compression layer, we use ontologically-defined frame relations to extract semantically similar frames from the predicted frame of the first layer. These semantically similar frames guide the compression layer of the model to infer appropriate abstract frames.

3.2 Input and Output

The input to our model is event sequences. Each sequence is defined by M event tuples $(x_1, x_2, ...x_M)$. For comparability (Weber et al., 2018; Rezaee and Ferraro, 2021), we represented each event as a tuple x_m of four lexical words: a predicate, a subject, an object, and an optional event modifier. We assume an event tuple can be associated with a more general semantic frame. For example, in Fig. 2, the first event ("work farmers in field") can be linked to the FrameNet AGRICULTURE frame. We assume that each event *can* be linked but do not require this. Some frames might be masked, subject to a fixable observation probability. This allows us to test how our model behaves when semantic data may be missing or incorrect (due to, e.g., an extraction error); in Fig. 2, this can be seen for the event "track animals in forest" event, where a potential corresponding frame—"Hunting_Scenario"—is masked. This results in a corresponding sequence of (partially) observed frames $(f_1^*, f_2^*, ...f_M^*)$. The base layer uses these event tuples (x_i) to softly predict the frames (f_i) and then reconstruct the input sequence based upon those inferences. To capture additional semantic knowledge, both in traning and testing, we query FrameNet to extract more abstract frames (h_i) for the predicted frames from the base layer, such as "Attempt_obtain_food_scenario." The compression layer uses that abstract frame h_i with the original event frames f_i to softly group the events; for additional training signal, the compression layer is also trained to reconstruct the original event sequence.

Encoder The base layer embeds each token in the input event sequence, while, by default, the compression layer embeds each predicted frame from the base layer. An attention module is used to find the important parts of event sequences during prediction of frames. As our experiments validate, the encoder can be flexible, e.g., a bi-GRU or a Transformer-based large language model.

Decoder This is a standard auto-regressive model that generates tokens of an event sequence from left to right. Unless otherwise specified, the predicted frame embeddings are given as input to the decoder. See App. A.1 for additional details.

3.3 Hierarchical Model

We use two layers of an encoder-decoder: (i) a base layer (f_is in Fig. 2) and (ii) a compression layer (h_js in Fig. 2). The base layer is responsible for encoding the input event sequence into a sequence of semantic frames, while the compression layer is responsible for re-encoding the base layer's semantic frames into more abstract representations. In Fig. 2, the base layer must infer "Agriculture" & "Hunting_Scenario" from the input and observed frames; the compression layer must associate those frames with "Attempt_obtain_food_scenario." Our model is extendable to an arbitrary number of compression layers. Experiments with multiple compression layers showed that a single compression layer was sufficient for strong performance.

Given our encoder-decoder setup, inferring frame values means sampling a discrete random variable within a neural network. This must be done at both the base and compression layers. To do so, we sample frames from an ancestral Gumbel-

Softmax distribution (Jang et al., 2016; Rezaee and Ferraro, 2021): each sampled frame f_i depends on the previously sampled frame f_{i-1} and an attention weighted embedding of that layer's encoder representation. Due to space, we refer the reader to Rezaee and Ferraro (2021).

Base Layer The base layer encodes the event sequences in the same number of latent variables with the guidance of the observed frames. On the base layer, partially observed frames are fed to the model. These frames depend on the observation probability; e.g., 40% observed frames mean that 60% of the event frames will be masked, and the remaining 40% would be observable by the model as guidance. This masking, which we formalize as part of our experiments, reflects the fact that we may not always have access to sufficient semantic knowledge. To guide the base layer, a one-hot encoding of the observed frames is "injected" (added to the Gumbel-Softmax parameters), as done by Rezaee and Ferraro (2021). The number of frames is the same as the number of event sequences, so one frame for each node is passed.

Compression Layer Rezaee and Ferraro (2021) showed that providing some frame injection guidance helps learning. The compression layer aims to provide guidance to the modeling through fewer, more abstract semantic frames. However, while this is possible for the base layer, where we assume every event tuple *could* have a frame, we do not assume this for the compression layer. This in part is reflective of the lack of annotated training samples for some of these more abstract frames (see §2.3), limited beyond-sentence frame extraction tools, and our own motivation to not require beyond-sentence annotation or extraction tools.

To provide guidance, but prevent reliance on potentially missing auxiliary semantic knowledge, we extract the *inferred* frames from the base layer with the external frame ontology (rather than whatever frames may have been provided to the model). For each inferred frame f_i, we extract possible abstract frames using the FrameNet relations defined for it. E.g., since there is a frame relation between AGRICULTURE and ATTEMPT_OBTAIN_FOOD_SCENARIO, if f_i is AGRICULTURE, ATTEMPT_OBTAIN_FOOD_SCENARIO may be an abstract frame. In the case of multiple abstract frames, one single frame is chosen randomly. A special frame token (not in FrameNet) is passed if no related frames can be extracted. Each compression node h_j has an attention module, attending over the base layer's inferred frames $f_1, \ldots, f_M$, helping capture ontological hierarchy.

While the compression layer can serve as an event model in its own right (due its own decoder), its primary purposes are to help **capture the ontological hierarchy and provide feedback to the base layer**. It does this directly (predict the extracted abstract frames, given the base layer's inferred frames as input), and via its decoder.

Guidance for Abstract Frames To guide the compression layer to learn more abstract frames and help the base layer generalize, we injected the FrameNet-defined parents of the frames predicted from the base layer. E.g., if the base layer prediction is "Temporary_Stay" and a related frame is "Visiting," we inject both to the compression layer. In contrast to existing work relying on single samples, early experiments showed that averaging two Gumbel-Softmax samples yielded better results.

3.4 Training

During training, input is passed to the base layer with partially observed frames depending on the observation probability. The first layer encoder encodes the input sequence with the guidance of the partially observed frames to generate a latent variable representation (f_i). This predicted latent variable (f_i) is then passed through the decoder to regenerate text. The predicted frames from the first layer and their parent frames are passed to the second layer encoder; it then encodes it to fewer numbers of latent variables, (h_j) which is used in the decoder. Loss is computed at both layers.

We employ a linear combination of three different loss functions: the reconstruction loss, the KL divergence loss, and a frame classification loss. The reconstruction loss is used to generate the input event sequence based on the inferred latent variables from each layer. The KL divergence loss calculates the KL divergence between the prior and variational distributions for each layer. Finally, the frame classification loss guides the base layer to accurately classify the observed frames. See App. A.3 for a full formulation of our loss.

4 Experimental Setup

We describe the dataset, then baselines (§4.1), we used for our core experiments. We explored the effectiveness of latent parent frames (§5.1) and frame relations (§5.2). We show how our model accounts

for missing events (§5.3). To further show the effectiveness of our model, we show how to extend our approach to provide effective representations for event similarity tasks (§5.4). We provide supplementary results and experiments in the appendix.

Dataset We used a part of the Concretely Annotated Wikipedia dataset (Ferraro et al., 2014), which is a version of English Wikipedia that provides automatically produced FrameNet semantic frame parses to enable easier subsequent examination of semantic frames. This has existing splits of training (457k), validation (16k), and test (21k) event sequences, where each training sequence has at least one extracted frame. For comparability with past approaches, we truncated documents to the first 5 events. We used a vocabulary size of 40k for event sequences (predicates and arguments) and the 500 most common semantic frames, which is consistent with prior work and has more than 99% coverage of automatically extracted frame types.

4.1 Implementation and Baselines

We use five latent variables in the base layer and three in the compression layer; these values were determined in early dev experiments. We represent the probability of observing an event's frame on the base layer with an observation probability ϵ. With ϵ likelihood, an event's frame will be observed, and with $(1-\epsilon)$ probability, an event's frame will be masked. This is meant to emulate how sufficiently accurate, extractable semantic knowledge may not always be available. This ϵ was fixed prior to training each model. Frames are **only observed during training, and never during evaluation**. More implementation details, including specific hyperparameter values and architectural decisions, are in App. A.2. We present extensive ablation experiments in App. C. These experiments provide further insight into our modeling decisions.

Baselines Most of our experiments (§ 5.1 to 5.3) compare our model with the existing methods: First, **HAQAE** (Weber et al., 2018), which employs a single layer, chain-based method for hierarchical modeling. It is designed purely as an unsupervised approach, and so we cannot provide frame guidance to it. We retrained this model on our event sequences. Second, **SSDVAE** (Rezaee and Ferraro, 2021): this is most similar to ours and effectively just the base layer. For fairness, we use the same hidden state size and pre-trained embeddings across our models and baselines.

Model	ϵ	Perplexity (↓)	INC Score (↑)
HAQAE	-	21.38 ± 0.25	24.88 ± 1.35
SSDVAE	0.9	19.84 ± 0.52	35.56 ± 1.70
ours: inf. frame	0.9	**19.39 ± 0.3**	41.35 ± 4.25
SSDVAE	0.7	21.19 ± 0.76	*39.08 ± 1.55*
ours: inf. frame	0.7	*20.26 ± 1.36*	35.86 ± 3.43
SSDVAE	0.5	31.11 ± 0.85	*40.18 ± 0.90*
ours: inf. frame	0.5	*22.16 ± 1.62*	37.3 ± 3.33
SSDVAE	0.4	33.12 ± 0.54	*47.88 ± 3.59*
ours: inf. frame	0.4	*24.02 ± 1.28*	43.25 ± 4.97
SSDVAE	0.2	33.31 ± 0.63	44.38 ± 2.10
ours: inf. frame	0.2	*30.15 ± 2.73*	**49.53 ± 1.56**

Table 1: Perplexity (lower is better) and Wikipedia Inverse Narrative Cloze Score (higher is better) for test data. Per observation probability (ϵ), the best is in *italic* form. The best overall is **bold** form. See §5.1.

5 Result and Discussion

We compute standard event modeling metrics: perplexity, to measure how well the model can predict the *next* event, and inverse narrative cloze (INC) score (Weber et al., 2018). In INC, a single seed event is given, and the model must select what the next five events are to follow it. The model is given six choices (giving random performance accuracy of 16.7%). Both have been used by our baselines and allow us to assess the effectiveness of our model. We average results over four runs with different seeds, unless otherwise specified.

5.1 Is Frame Inheritance Sufficient?

We first investigate whether frame inheritance is sufficient for learning our hierarchical model. We report the inferred frame variant previously described: the base layer first infers the latent frames; then we extract the parents of those inferred frames; and we then inject both these parent frames and base layer predicted frames in the compression layer. The compression layer is dependent on the inferred frames, rather than lexical signal. Results are in Table 1 (supplemental results in Tables 6 and 7 in the appendix). We also experimented with a lexical variant, where the input to the compression layer is an embedding of the original input event tuple rather than the inferred frames. Due to space constraints, these detailed comparisons are in App. B.1. The compression layer alone has suboptimal performance on both lexical and inferred frame models, but the signal from compression layer helped the base layer to achieve better performance. Both SSDVAE and HAQAE (no compression layer) did worse for all observation probabilities. This shows the inferred frames and semantic relations from the

base layer are important for hierarchical modeling.[2]

Our model's base layer perplexity consistently outperformed the other models. Additionally, we see that our approach is better able to handle lower supervision than SSDVAE: as the observation probability decreases (fewer observed semantic frames), perplexity increases drastically for SSDVAE. In contrast, if we look at the "ours: inf. frames" perplexity, we see that any performance degradation in our model is less severe, and that in all cases our approach still outperforms the previous SOTA results. This shows the effectiveness of the compression layer in guiding the base layer reconstruction, even with limited semantic observation.

Looking at INC, with either a lot ($\epsilon = 0.9$) or a little ($\epsilon = 0.2$) of semantic observations, our approach outperforms the existing approaches, demonstrating the ability to model longer event sequences. The best overall INC performance occurs with our hierarchical model with a low amount of supervision. This is a good result, as it suggests our model can make use of limited semantic extractions and still provide effective long-range modeling. When some, but not necessarily most, of the frames may be observed, the non-hierarchical SSDVAE approach provides strong performance. This suggests that while frame inheritance (e.g., *IS-A* type relations) can be helpful for certain elements of hierarchical event modeling, it is not sufficient. However, as we will see in the next section, more considered use of semantic relations defined in FrameNet can drastically boost our model's performance, surpassing SSDVAE.

5.2 Relations Beyond Inheritance

We have shown that inheritance relations are helpful but not sufficient. As FrameNet reflects other relations, like causation, (temporal) ordering, and multiple forms of containment/composition, we explore whether six different frame relations significantly affect the predictive abilities of our model.

We also consider two special cases: first, whether different types of relations are complementary by grouping these select relations.[3] We refer to this as *grouping* in Table 2. Second, whether the compositional "scenario" frames in FrameNet provide a strong signal (*scenario-only* in Table 2). In FrameNet, frames that introduce a broader, abstract concept rather than an isolated one can be labeled as a "scenario" frame: e.g., *COMMERCE_SCENARIO* consists of buying, selling, business, having an agreement, and so on. For this, we only extracted an abstract frame for the compression layer if it was labeled as a "scenario."

Model	Frame Relation	ϵ	Next Event Pred. (Perplexity)	Event Sequence Pred. (Wiki INC Accuracy)
HAQAE	-	-	21.38 ± 0.25	24.88 ± 1.35
SSDVAE	-	0.9	19.84 ± 0.52	35.56 ± 1.70
ours	Inheritance	0.9	19.39 ± 0.53	41.35 ± 4.25
	Using		19.39 ± 0.51	*43.23 ± 2.51*
	Precedes		19.57 ± 0.58	41.43 ± 3.02
	Causative_of		19.42 ± 0.57	41.38 ± 2.23
	Inchoative_of		19.28 ± 0.32	41.35 ± 3.47
	Perspective_on		19.76 ± 0.97	40.53 ± 2.04
	Subframe		18.91 ± 0.15	40.35 ± 2.91
	grouping		19.44 ± 0.5	40.76 ± 2.86
	scenario-only		***18.81 ± 0.5***	42.29 ± 2.86
SSDVAE	-	0.2	33.31 ± 0.63	44.38 ± 2.10
ours	Inheritance	0.2	*30.15 ± 2.73*	49.53 ± 1.56
	Using		31.37 ± 2.08	49.72 ± 1.73
	Precedes		32.62 ± 1.65	47.92 ± 2.25
	Causative_of		31.82 ± 3	***49.85 ± 0.84***
	Inchoative_of		32.65 ± 1.4	48.03 ± 3.35
	Perspective_on		33.2 ± 1.47	47.85 ± 3.53
	Subframe		32.78 ± 2.09	47.88 ± 3.31
	grouping		28.17 ± 2.26	48.88 ± 1.37
	scenario-only		32.01 ± 0.7	48.1 ± 2.22

Table 2: Using frame relations beyond inheritance for the compression layer can lead to drastic improvements in both perplexity (lower is better) and Wikipedia Inverse Narrative Cloze Score (higher is better). See §5.2. For detailed result with all the layers, please refer to appendix (Apps. B.2 to B.4).

We trained separate models (with three random seeds) for each frame relation to explore the effect of individual frame relations on the result. We focus on higher ($\epsilon = 0.9$) and lower ($\epsilon = 0.2$) frame observation cases. Table 2 shows our main results, with detailed results in the appendix (Apps. B.2 to B.4). Lower observation ($\epsilon = 0.2$) is consistently better than the previous state-of-the-art on the base and overall versions. For $\epsilon = 0.9$, base layer performance is generally improved. This reaffirms our previous results that even with limited semantic guidance, the compression layer provides valuable feedback to the base layer.

The results for the two special relations in Table 2 (*grouping* and *scenario-only*) are consistent with our previous results—our approach outperforms the state-of-the-art result. Neither grouping nor the scenario-only variant provides large additional benefit beyond the individual frames in that group. Given this and the small variation in base layer performance depending on what frame relations we use, these results suggest that the ex-

[2] In particular, Fig. 4 in the appendix shows how the compression layer can demonstrate its own generative capabilities, in addition to providing supervisory signal to the base layer.

[3] We aggregate frames connected via the Inheritance, Using, Precedes, Causative_of, Inchoative_of, and Subframe relations. We selected these given their direct connections to well-studied relationships across event semantics.

Model	ϵ	Perplexity (Masked Test Data)		
		Base Alone	Compression Alone	Base+Compr.
SSDVAE		152.44 ± 3.45	-	-
grp	0.9	*61.1 ± 1.83*	94.76 ± 1.96	76.08 ± 0.76
scn		63.48 ± 4.43	80.94 ± 7.44	71.6 ± 4.12
SSDVAE		163.08 ± 4.52	-	-
grp	0.7	63.5 ± 3.49	86.23 ± 0.7	73.98 ± 2.04
scn		***60.06 ± 1.68***	78.36 ± 4.52	68.58 ± 2.3
SSDVAE		182.63 ± 6.11	-	-
grp	0.5	79.74 ± 1.79	83.81 ± 0.96	81.75 ± 1.13
scn		*76.01 ± 5.56*	78.7 ± 1.63	77.33 ± 3.65
SSDVAE		201.55 ± 4.1	-	-
grp	0.4	84.17 ± 4.45	81.49 ± 0.14	82.8 ± 2.13
scn		*73.77 ± 7.87*	80 ± 1.89	76.77 ± 4.89
SSDVAE		212.93 ± 2.54	-	-
grp	0.2	89.73 ± 4.67	77.32 ± 0.72	83.28 ± 2.38
scn		83.86 ± 2.74	81.2 ± 1.17	*82.52 ± 1.93*

Table 3: Perplexity (lower is better) for the grouped and scenario-based models in the scenario-masked evaluation. For each ϵ, the best score is *italicized*. Best overall is **bold**. These results indicate how our approach can make use of related frames to better model sequences involving missing events. See §5.3.

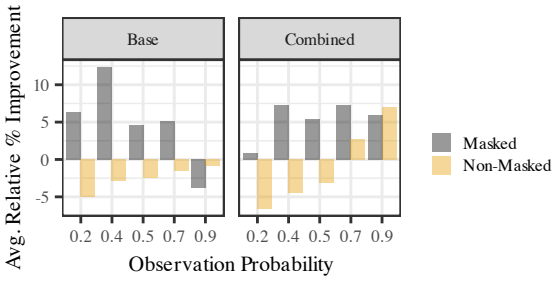

Figure 3: Relative perplexity *improvement* of the scenario-based model vs. the grouped model; higher is better. The scenario model improves across observation levels when important events are missing.

istence of broader assocations that these relations enable are very helpful. This would suggest that semantically-aware event modeling could benefit from broader semantic resource coverage, with future work examining how best to encode the semantics of *any particular* relation.

5.3 Predicting Missing Events

Previously, we have looked at how using the observation probability can help us mask frames and semi-supervised learning. In this experiment, we examine the robustness of our model with respect to missing events in an input sequence along with the frame masking depending on observation probability. We first identify sequences (in our training, dev, and test data) where two events have different frames f_i and f_j that are contained within the same scenario frame. We train normally, but to evaluate, we remove an event e_j associated with a scenario-connected frame f_j from the input. Given this impoverished input, we require the model to generate the full, unmodified sequence. By construction, the missing event is not a randomly missing event: it is, according to the semantic ontology, *semantically related to another event in that sequence*. To compare our model with SSDVAE, we have trained SSDVAE with the same data and evaluated with the same masked input and full event regeneration.

Given their strong performance, we examine the grouped and scenario-based models. Results, averaged across three seeds, are in Table 3: *grp* is the model with a group of FrameNet relations, *scn* is the model with scenario sub-frames and *SSDVAE* is the SSDVAE model with different evaluation. To show the consistent benefits of our approach, we report results computed just from the base decoder, just from the compression decoder, and from a score combined from both the base and compression decoders. When an important event is masked, the scenario-based model nearly always outperforms the grouped model across observation levels. Our model can leverage training time scenario-related frame associations to better predict a missing event. Also, for all observation probabilities, both of our model's (*grp* & *scn*) individual and combined layer outperform SSDVAE. We suspect this is because SSDVAE does not have a hierarchical abstraction mechanism, so when one event is not present, the related frame is also missing. This shows the capability of the hierarchical structure of our model to understand and encapsulate the abstract meaning of an event sequence.

It is not surprising that the base layer, with more feedback during training and greater representational capacity, is a better language model than the compression layer on its own. Still, the compression layer provides active benefits: we summarize the relative improvement of the scenario-based model over the grouped model in Fig. 3. We compute this from just the base layer, or from both the base and compression layers. A positive number means that the scenario-based model was better able to (re)generate a full event sequence compared to the grouped model. Except for very high observation probability on the base layer, the **scenario-based model consistently outperformed the grouped one when semantically-relevant events were missing**. The grouped model, which covers multiple frame relations, can better model sequences when events are

not missing. While this may seem intuitive, notice how using the compression layer is able to reverse this pattern and let the scenario-based model outperform the grouped one, highlighting the benefit that the compression layer can bring.

5.4 Improved Event Similarity

We have shown that both structural and semantic hierarchy can be beneficial when predicting the next event in a sequence, "rolling out" a longer sequence from an initial seed, and accounting for semantically missing events. In our final experiment, we use the latent frame representation to improve the overall event representation. We evaluate on three similarity datasets, comparing to the state-of-the-art (Gao et al., 2022). In two of the datasets, there are two event pairs and the task is to determine which pair is more similar (measured by accuracy); the third involves scalar human assessment scores for how related two events are (Spearman correlation). Data are only for evaluation, and all training is done as "pre-training." As such, our experiments demonstrate the ability to capture semantic information in our latent variable representation, and to perform in an evaluation-only (zero-shot) prediction of semantically-related events.

Gao et al. (2022) presents SWCC, a **s**imultaneous, **w**eakly supervised, **c**ontrastive learning and **c**lustering framework for event representation learning. They combine a clustering loss with the popular contrastive learning approach of InfoNCE (Oord et al., 2018). Every "query" point x (an event tuple) has positive (similar) instances $z_1, ..., z_R$, and negative (dissimilar) instances $z_{R+1}, ..., z_S$. Using a temperature-annealed similarity function on model-computed embeddings, e.g., cosine similarity on embeddings from a LLM, a probability distribution is computed over the positive and negatives (conditioned on the query). Average cross-entropy is optimized to predict the positive vs. negative instances.

This contrastive loss nicely augments our model's existing training objective from §3.4. We pre-train our hierarchical model on the same partially observable frame-annotated data from §4, using that model to extract a representation for an event, and computing the cosine similarity between two representations. We form a representation by concatenating the decoder's final token embedding and the latent frames from the compression layer. To prevent frame representations overfitting to the predicates, rather than arguments, we applied a predicate-specific dropout of 70% on the encoder. Our hierarchical model provides a straightforward way to adopt contrastive loss; this hierarchical nature is not explicit in SSDVAE or HAQUE. Adapting these approaches to the contrastive learning setup is beyond the scope of our work.

Model	Hard Similarity (Accuracy %)		Transitive Score
	Original	Extended	Similarity
SWCC (16)	78.91 ± 1.31	69.2 ± 0.93	0.82 ± 0
SWCC (256)	81.09 ± 0.43	72.55 ± 1.53	**0.82 ± 0**
Ours	**83.26 ± 2.29**	**78.63 ± 2.95**	0.77 ± 0.04

Table 4: Evaluation on Similarity Tasks. SWCC (256) are Gao et al.'s reported results, using a batch size of 256. Given the importance that batch size can have with contrastive learning, we ran Gao et al.'s model with a batch size 16 (the same batch size of our model). We report this as SWCC (16). See §5.4.

	Training Variant	Hard Similarity (Accuracy %)		Transitive Score
		Original	Extended	Similarity
Ours (16)	Contrastive + LM	83.26 ± 2.29	78.63 ± 2.95	0.77 ± 0.04
	Contrastive only	67.18 ± 1.79	72.75 ± 2.06	0.72 ± 0.02
	LM only	67.83 ± 14.39	62.15 ± 16.52	0.56 ± 0.04
SWCC (16)	Contrastive + MLM	78.91 ± 1.31	69.2 ± 0.93	0.82 ± 0
	Contrastive only	78.48 ± 0.83	67.33 ± 0.19	0.78 ± 0.05
	MLM only	25.87 ± 1.31	16.78 ± 0.7	0.55 ± 0.04

Table 5: Ablation study of our model and SWCC.

Our results are in Table 4. We have run SWCC with a batch size of 16, which is the same as ours. Our model surpasses SWCC on two of the tasks, showing it is not only capable of event language modeling but also capable of generating better event representations. We have also run an ablation study on SWCC and our model; the results are on Table 5. The results show that neither contrastive nor LM/MLM loss are as strong as both together. We see that the LM component in our approach is important to overall performance.

6 Conclusion

We have presented a hierarchical event model that accounts for both structural and ontological hierarchy across an event sequence. We use automatically extracted semantic frames to guide the first level of concept, and then use FrameNet relations to guide abstraction and generalization. We showed improvements across multiple tasks and evaluation measures within event modeling. We showed improvements in next event prediction, longer range event prediction, missing event regeneration, and event similarity. We believe that future work can use this abstraction concept for summarization, topic modeling, or other downstream tasks.

7 Limitations

Our approach enables modeling observed event sequences through the lens of a structured semantic ontology. Though our models have shown superior performance to leverage event frames, they still suffer from the bottleneck of the information passed to the compression layer. Additionally, while these resources do exist, their coverage is not universal, and have historically been developed for English. Our experiments reflect this.

While the observance of frames is not, strictly speaking, a requirement of our model, our experiments focused on those cases when such an ontology is available during training.

Throughout our experiments, we use pretrained models/embeddings. We do not attempt to control or mitigate any biases these may exhibit or propagate.

Our work does not involve human subjects research, data annotation, or representation/analysis of potentially sensitive characteristics. As such, while we believe the direct *potential risks* of our approach are minimal we acknowledge that the joint use of pretrained models and structured semantic ontologies could result in undesired or biased semantic associations.

Acknowledgments

We would like to thank the anonymous reviewers for their comments, questions, and suggestions. This material is based in part upon work supported by the National Science Foundation under Grant No. IIS-2024878. Some experiments were conducted on the UMBC HPCF, supported by the National Science Foundation under Grant No. CNS-1920079. This material is also based on research that is in part supported by the Army Research Laboratory, Grant No. W911NF2120076, and by the Air Force Research Laboratory (AFRL), DARPA, for the KAIROS program under agreement number FA8750-19-2-1003. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of the Air Force Research Laboratory (AFRL), DARPA, or the U.S. Government.

References

Roni Ben Aharon, Idan Szpektor, and Ido Dagan. 2010. Generating entailment rules from framenet. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 241–246.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *In Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. Citeseer.

Yonatan Bisk, Jan Buys, Karl Pichotta, and Yejin Choi. 2019. Benchmarking hierarchical script knowledge. 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 4077–4085.

Teresa Botschen, Hatem Mousselly-Sergieh, and Iryna Gurevych. 2017. Prediction of frame-to-frame relations in the FrameNet hierarchy with frame embeddings. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*.

Muhao Chen, Hongming Zhang, Haoyu Wang, and Dan Roth. 2020. What are you trying to do? semantic typing of event processes. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 531–542.

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. Probabilistic frame induction. In *NAACL*.

Alan Darmasaputra Chowanda, Albert Richard Sanyoto, Derwin Suhartono, and Criscentia Jessica Setiadi. 2017. Automatic debate text summarization in online debate forum. *Procedia computer science*, 116:11–19.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 16–23, Madrid, Spain. Association for Computational Linguistics.

Dipanjan Das, Desai Chen, André FT Martins, Nathan Schneider, and Noah A Smith. 2014. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56.

Xinya Du, Alexander Rush, and Claire Cardie. 2021. GRIT: Generative role-filler transformers for

document-level event entity extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online. Association for Computational Linguistics.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Carl Edwards and Heng Ji. 2022. Semi-supervised new event type induction and description via contrastive loss-enforced batch attention. *arXiv preprint arXiv:2202.05943*.

Francis Ferraro, Adam Poliak, Ryan Cotterell, and Benjamin Van Durme. 2017. Frame-based continuous lexical semantics through exponential family tensor factorization and semantic proto-roles. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 97–103, Vancouver, Canada. Association for Computational Linguistics.

Francis Ferraro, Max Thomas, Matthew R Gormley, Travis Wolfe, Craig Harman, and Benjamin Van Durme. 2014. Concretely annotated corpora. In *4th Workshop on Automated Knowledge Base Construction (AKBC)*.

Francis Ferraro and Benjamin Van Durme. 2016. A Unified Bayesian Model of Scripts, Frames and Language. In *Proceedings of the 30th Conference on Artificial Intelligence (AAAI)*, pages 2601–2607, Phoenix, Arizona. Association for the Advancement of Artificial Intelligence.

Jun Gao, Wei Wang, Changlong Yu, Huan Zhao, Wilfred Ng, and Ruifeng Xu. 2022. Improving event representation via simultaneous weakly supervised contrastive learning and clustering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3036–3049, Dublin, Ireland. Association for Computational Linguistics.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Yong Guan, Shaoru Guo, Ru Li, Xiaoli Li, and Hu Zhang. 2021. Frame semantics guided network for abstractive sentence summarization. *Knowledge-Based Systems*, 221:106973.

Xu Han, Tao Lv, Zhirui Hu, Xinyan Wang, and Cong Wang. 2016. Text summarization using framenet-based semantic graph model. *Scientific Programming*, 2016:5.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, pages 57–60. Association for Computational Linguistics.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. DEGREE: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Chieh-Yang Huang and Ting-Hao Huang. 2021. Semantic frame forecast. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

James Y. Huang, Bangzheng Li, Jiashu Xu, and Muhao Chen. 2022. Unified semantic typing with meaningful label inference. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

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.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Tim O'Gorman, Kristin Wright-Bettner, and Martha Palmer. 2016. Richer event description: Integrating event coreference with temporal, causal and bridging annotation. In *Proceedings of the 2nd Workshop on Computing News Storylines (CNS 2016)*, pages 47–56.

Ankur Padia, Francis Ferraro, and Tim Finin. 2018. Surface: semantically rich fact validation with explanations. *arXiv preprint arXiv:1810.13223*.

Sveva Pepe, Edoardo Barba, Rexhina Blloshmi, and Roberto Navigli. 2022. Steps: Semantic typing of event processes with a sequence-to-sequence approach.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia. Association for Computational Linguistics.

Mehdi Rezaee and Francis Ferraro. 2021. Event representation with sequential, semi-supervised discrete variables. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4701–4716, Online. Association for Computational Linguistics.

Karin Kipper Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, University of Pennsylvania.

Beth Sundheim. 1992. Proceedings of the fourth message understanding conference (MUC-4).

Beth Sundheim. 1996. Overview of results of the MUC-6 evaluation. In *Proceedings of a Workshop held at Vienna, Virginia: May 6-8, 1996*.

Noah Weber, Leena Shekhar, Niranjan Balasubramanian, and Nathanael Chambers. 2018. Hierarchical quantized representations for script generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3783–3792.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020. Reasoning about goals, steps, and temporal ordering with WikiHow. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

A Additional Model and Implementation Details

A.1 Model Details

For our input data, events are separated by a <*TUP*> token, and in case of missing values in an event frame, is replaced with a special <*NOFRAME*> token.

As mentioned in the main paper, like any auto-regressive model, previously generated decoder output and previous input texts are given as input to the decoder. An attention module is used to find the important words from the given latent embeddings predicted by encoder. Each layer tries to reconstruct the input text, and loss was generated individually for each layer, which then accumulated and back-propagated through the whole model, updating the model parameters.

A.2 Implementation Details

The values of γ_1 and γ_2 are set to 0.1 by experimenting on the validation set. 2 Gumbel-softmax samples are used to average the encoder. We use the Adam (Kingma and Ba, 2014) optimizer with a learning rate of 0.001. A batch size of 64 has been used with a gradient accumulation of 8. Early stopping has been used with patience of 10 on the validation perplexity score.

For comparability, our core event modeling results use recurrent encoders and decoders. We use pretrained Glove-300 embeddings to represent each lexical item in an event tuple. An embedding size of 500 has been used for frame embeddings. Two layers of bidirectional GRU have been used for the encoder, and two layers of uni-directional GRU have been used for the decoder. Both are used with 512 hidden sizes. Gradient clipping of 5.0 has been used to prevent gradient exploding. 0.5 has been used as the Gumbel-softmax temperature.

Similarly, our experiments involving event similarity (§5.4) use BART (Lewis et al., 2019) as our encoder and decoder module.

Across our experiments, we have used NVIDIA RTX 2080Ti or NVIDIA RTX 6000 for training. It takes around 16 hours to train with our current batch size on our dataset.

A.3 Loss Formulation

In constructing our training loss function, we take inspiration from the methodology outlined in the study conducted by (Rezaee and Ferraro, 2021). However, our model differs in that it incorporates two hidden layers, as opposed to the single latent layer utilized in the aforementioned study. Each layer we calculate the loss for both layers individually. This is done by allowing each layer j, to reconstruct the input text using its own latent

variables, L_{r_j}. To prevent overfitting, we incorporate KL terms in our loss function denoted as $\mathcal{L}_{\text{KL}_j}$. Additionally, for the base layer we include a classification term, designated as $\mathcal{L}_c$.

$$\begin{aligned}\mathcal{L} = & \underbrace{\alpha_1 * \mathcal{L}_{r_1} + \alpha_2 * \mathcal{L}_{r_2}}_{\text{Text Reconstruction}} \\ & + \underbrace{\beta_1 * \mathcal{L}_{\text{KL}_1} + \beta_2 * \mathcal{L}_{\text{KL}_2}}_{\text{Regularization}} \\ & + \underbrace{\gamma * \mathcal{L}_c}_{\text{Observed Frame Classification}}.\end{aligned} \tag{1}$$

The reconstruction and KL losses depend on the random variables inferred at each level: for the base level ($j = 1$), the losses depends on the frames sampled at the base level $f_1, \ldots, f_n$, while the compression losses ($j = 2$) depend on $h_1, \ldots, h_M$). Our latent variable model learns a variational distribution q, from which it can infer appropriate values for f_i and h_j. With this, we compute

$$\mathcal{L}_{r_1} = \mathbb{E}_{q(f_1,\ldots,f_N)}[\log p(x|f_1, \ldots, f_N)] \tag{2}$$

$$\mathcal{L}_{r_2} = \mathbb{E}_{q(h_1,\ldots,h_M)}[\log p(x|h_1, \ldots, h_M)] \tag{3}$$

$$\mathcal{L}_{\text{KL}_1} = \mathbb{E}_{q(f_1,\ldots,f_N)}[\log p(f_1, \ldots, f_N)] \tag{4}$$

$$\mathcal{L}_{\text{KL}_2} = \mathbb{E}_{q(h_1,\ldots,h_M)}[\log p(h_1, \ldots, h_M)] \tag{5}$$

$$\mathcal{L}_c = - \sum_{i=1:f_i^* \text{ is obs.}}^{N} \log q(f_i^*|f_{i-1}). \tag{6}$$

In $\mathcal{L}_c$, note that f_i^* represents the correct value of the ith frame. The reconstruction and frame classification losses can be computed via a cross-entropy loss (per output token for the reconstruction losses, and per predicted frame in the frame classification loss).

B Additional Results

B.1 Is Frame Inheritance Sufficient?

The detailed results for the experiment described in §5.1 are reported in Table 6 (Perplexity Score) and Table 7 (INC).

Detailed per-layer perplexity is reported in Table 6, augmenting the results in Table 1. Our model's base layer perplexity consistently outperformed the other models. However, the perplexity of the compression layer was higher. This suggests that while incorporating hierarchical layers or knowledge may not be sufficient for generating the event sequence, it provides useful, less-than-full supervised feedback to the base layer.

Model	ϵ	Perplexity (Test Data) Base	Compression	Total
HAQAE	-	-	-	21.38 ± 0.25
SSDVAE		-	-	19.84 ± 0.52
ours: inf. frame	0.9	*19.39 ± 0.3*	26.52 ± 0.55	22.68 ± 0.41
ours: lexical		***19.12 ± 0.53***	31.43 ± 1.1	24.51 ± 0.39
SSDVAE		-	-	21.19 ± 0.76
ours: inf. frame	0.7	*20.26 ± 1.36*	27.45 ± 0.5	23.57 ± 0.84
ours: lexical		*21.52 ± 1.48*	35.19 ± 0.95	27.5 ± 0.93
SSDVAE		-	-	31.11 ± 0.85
ours: inf. frame	0.5	*22.16 ± 1.62*	32.59 ± 2.86	26.62 ± 2.13
ours: lexical		*25.02 ± 1.31*	39.44 ± 0.44	31.41 ± 0.77
SSDVAE		-	-	33.12 ± 0.54
ours: inf. frame	0.4	*24.02 ± 1.28*	32.82 ± 1.44	28.07 ± 1.24
ours: lexical		*27.06 ± 0.94*	40.46 ± 2.74	33.05 ± 0.56
SSDVAE		-	-	33.31 ± 0.63
ours: inf. frame	0.2	*30.15 ± 2.73*	34.81 ± 2.81	32.84 ± 1.84
ours: lexical		*33.6 ± 1.84*	44.64 ± 1.44	38.72 ± 1.59

Table 6: Per-word perplexity for test data (lower is better). For each observation probability (ϵ), the best perplexity is in *italic* form. The best of all of them is **bold** form. See App. B.1

For INC, we look to the lexical variant, where our model's base layer outperforms the previous result with having the best of all the observation probabilities. However, the results for the compression layer underperformed the inferred variant, indicating that incorporating lexical signals may have a negative impact on the performance of the generation model. Overall, this suggests that the **inferred frames and ontological relations from the base layer are important for hierarchical modeling**.

We have reported an average change in the INC score of the base layer over the combined layer on Fig. 4. The gray and orange bars represent the two variants: inferred frames and lexical signal, respectively. Each bar is the average of the score change from the combined layer to the base layer (combined layer score – base layer score). Here, a negative score means that the base layer is better than the combined one. This figure shows if the use of compression layer has a positive impact on the INC score or not. First, for the inferred frames, the addition of a compression layer has improved the INC score by an effective margin on the base layer. This shows that the semantic frames have helped the model's base layer to understand the process better. On the other hand, for the lexical signal, the combined layer has a better INC score. This shows that having the lexical signal on the compression layer has a better and equal effect on both layers. In conclusion, the addition of a compression layer improves the model's capability of understanding event sequences and generalizing.

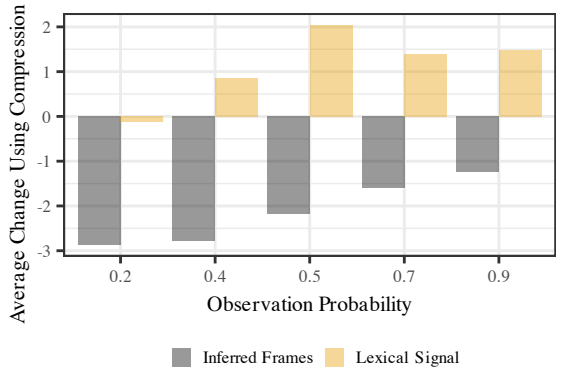

Figure 4: Average Change in INC score from combined layer to base layer, where a negative score means the base layer was better than the combined one and vice versa. The gray and orange bars indicate whether the input to the compression layer is inferred frames or lexical signal, respectively. In all cases, inferred frames has a better effect on base layer and lexical signal has improved combined layer's performance.

Model	ϵ	Wikipedia INC (Test Data) Base	Compression	Total
HAQAE	-	-	-	24.88 ± 1.35
SSDVAE		-	-	35.56 ± 1.70
ours: inf. frame	0.9	41.35 ± 4.25	27.25 ± 1.02	40.11 ± 1.88
ours: lexical		41.35 ± 3.19	35.41 ± 2.56	*42.83 ± 1.47*
SSDVAE		-	-	*39.08 ± 1.55*
ours: inf. frame	0.7	35.86 ± 3.43	26.31 ± 2.92	34.26 ± 3.43
ours: lexical		35.61 ± 4.72	32.68 ± 6.12	37.01 ± 6.59
SSDVAE		-	-	*40.18 ± 0.90*
ours: inf. frame	0.5	37.3 ± 3.33	23.61 ± 1.34	35.13 ± 3.01
ours: lexical		37.8 ± 3	37.11 ± 3.14	39.85 ± 3.01
SSDVAE		-	-	*47.88 ± 3.59*
ours: inf. frame	0.4	43.25 ± 4.97	23.65 ± 1.34	40.46 ± 4.71
ours: lexical		39.2 ± 1.23	34.79 ± 4.75	40.06 ± 2
SSDVAE		-	-	44.38 ± 2.10
ours: inf. frame	0.2	***49.53 ± 1.56***	25.15 ± 4.34	46.65 ± 1.55
ours: lexical		46.53 ± 2.84	37.55 ± 2.8	46.41 ± 3.71

Table 7: Wikipedia Inverse Narrative Cloze Score for test data (higher is better). For each observation probability (ϵ), the best score is in *italic* form. The best of all of them is **bold** form. See App. B.1

Model	Frame Relation	ϵ	Perplexity (Test Data) Base	Compression	Total
HAQAE	-	-	-	-	21.38 ± 0.25
SSDVAE	-	0.9	-	-	19.84 ± 0.52
ours	Using		19.39 ± 0.51	25.34 ± 0.22	22.16 ± 0.37
	Precedes		19.57 ± 0.58	25.83 ± 0.25	22.48 ± 0.25
	Metaphor		19.62 ± 0.75	25.21 ± 0.49	22.24 ± 0.63
	See_also		19.55 ± 0.72	25.71 ± 0.39	22.42 ± 0.54
	Causative_of		19.42 ± 0.57	25.75 ± 0.46	22.36 ± 0.53
	Inchoative_of		19.28 ± 0.32	26.01 ± 0.85	22.39 ± 0.52
	Perspective_on		19.76 ± 0.97	25.64 ± 0.57	22.5 ± 0.75
	Subframe		***18.91 ± 0.15***	26.03 ± 0.42	22.19 ± 0.27
	ReFraming_Mapping		19.56 ± 0.94	26.63 ± 1.81	22.81 ± 0.62
SSDVAE	-	0.2	-	-	33.31 ± 0.63
ours	Using		31.37 ± 2.08	38.55 ± 5.72	34.72 ± 3.23
	Precedes		32.62 ± 1.65	45.33 ± 0.74	38.45 ± 1.25
	Metaphor		32.92 ± 2.08	42.07 ± 5.83	37.18 ± 3.5
	See_also		31.83 ± 2.78	41.78 ± 5.55	36.44 ± 3.79
	Causative_of		31.82 ± 3	40.01 ± 6.23	35.67 ± 4.41
	Inchoative_of		32.65 ± 1.4	42.42 ± 3.55	37.21 ± 2.21
	Perspective_on		33.2 ± 1.47	44.18 ± 1.26	38.28 ± 0.34
	Subframe		32.78 ± 2.09	45.25 ± 0.7	38.51 ± 1.52
	ReFraming_Mapping		*31.34 ± 2.76*	36.57 ± 2.9	34.06 ± 3.15

Table 8: Per-word perplexity for test data (lower is better). For each observation probability (ϵ), the best perplexity is in *italic* form. The best of all of them is **bold** form. See App. B.2

B.2 The Effect of Individual Frame Relations

The detailed results for the experiment described in §5.2 are reported on Table 8 (Perplexity Score) and Table 9 (Wikipedia Inverse Narrative Score).

B.3 Are scenario subframes better than other frame properties?

The detailed results for the experiment reported in Table 2 are shown in Table 10 (Perplexity Score) and Table 11 (INC).

B.4 The Effect of Grouping Frame Properties

The previous section showed that performance of our model can be further improved by using targeted frame relations. Here, we investigate whether grouping of different frame relations could have a more significant impact on generalization.

Using §5.2, we identified six frame-relations as the most important ones: Inheritance, Using, Precedes, Causative_of, Inchoative_of, and Subframe. We used this group of frame relations to extract the parent frames from the predicted frames of the base layer. With their parent frames, these frames were passed to the compression layer to learn to associate the semantically similar frames.

Looking at the perplexity results (Table 12) of this experiment, we can see that the base layer outperforms both baselines across observation lavels. Additionally, while we see the intuitive result that higher levels of frame observation during training improves perplexity, we see the largest relative improvments for $\epsilon = 0.5$ and $\epsilon = 0.4$. This suggests that our hierarchical model is able to effectively leverage the semantic ontology, even when 40% of events do not have observed frames.

We see broadly similar patterns for inverse narrative cloze, with our approach outperforming both baselines. First, our performance is highest with the lowest observation level. Second, aside from when 90% of the events have observed frames, as ϵ decreases, so does our model's variance, while the previous state-of-the-art's increases. Taken together, these results suggest that our model is better able to use the provided semantic ontology and make better longer range predictions, even with lim-

Model	Frame Relation	ϵ	Wikipedia INC (Test Data) Base	Compression	Total
HAQAE	-	-	-	-	24.88 ± 1.35
SSDVAE	-	0.9	-	-	35.56 ± 1.70
ours	Using		43.23 ± 2.51	26.68 ± 0.63	40.92 ± 1.85
	Precedes		41.43 ± 3.02	26.38 ± 1.51	40.03 ± 1.66
	Metaphor		41.92 ± 3.93	24.22 ± 1.53	38.8 ± 2.17
	See_also		42.67 ± 1.49	27.08 ± 0.24	41.13 ± 0.81
	Causative_of		41.38 ± 2.23	26.3 ± 1.05	40.47 ± 1.79
	Inchoative_of		41.35 ± 3.47	26.67 ± 1.33	40 ± 2.34
	Perspective_on		40.53 ± 2.04	26.38 ± 0.67	39.55 ± 1.75
	Subframe		40.35 ± 2.91	25.7 ± 0.48	38.42 ± 2.32
	ReFraming_Mapping		*43.8 ± 4.02*	26.7 ± 1.21	42.15 ± 3.19
SSDVAE	-	0.2	-	-	44.38 ± 2.10
ours	Using		***49.72 ± 1.73***	21.77 ± 1.1	45.93 ± 1.62
	Precedes		47.92 ± 2.25	20.67 ± 0.29	42.72 ± 1.58
	Metaphor		47.25 ± 3.81	21.12 ± 0.95	42.77 ± 3.27
	See_also		47.77 ± 3.61	21.2 ± 1.15	43.72 ± 2.78
	Causative_of		49.85 ± 0.84	21.5 ± 2.41	45.45 ± 2.03
	Inchoative_of		48.03 ± 3.35	21 ± 0.74	43.95 ± 2.61
	Perspective_on		47.85 ± 3.53	20.42 ± 0.3	43.08 ± 3.12
	Subframe		47.88 ± 3.31	20.33 ± 0.52	42.38 ± 1.86
	ReFraming_Mapping		49.05 ± 1.54	22.23 ± 0.58	45.45 ± 0.44

Table 9: Wikipedia Inverse Narrative Cloze Score for test data (higher is better). For each observation probability (ϵ), the best score is in *italic* form. The best of all of them is **bold** form. See App. B.2

Model	ϵ	Perplexity (Test Data) Base	Compression	Total
HAQAE	-	-	-	21.38 ± 0.25
SSDVAE	0.9	-	-	19.84 ± 0.52
scenario-only		*18.81 ± 0.36*	25.61 ± 1.23	21.94 ± 0.5
SSDVAE	0.7	-	-	21.19 ± 0.76
scenario-only		***18.75 ± 0.3***	26.82 ± 0.47	22.42 ± 0.21
SSDVAE	0.5	-	-	31.11 ± 0.85
scenario-only		*23.79 ± 1.29*	31.43 ± 7.44	28.7 ± 2.04
SSDVAE	0.4	-	-	33.12 ± 0.54
scenario-only		*25.54 ± 2.34*	36.87 ± 6.01	30.63 ± 3.52
SSDVAE	0.2	-	-	33.31 ± 0.63
scenario-only		*32.01 ± 0.7*	45.28 ± 0.7	38.07 ± 0.55

Table 10: Per-word perplexity for test data (lower is better). For each observation probability (ϵ), the best perplexity is in *italic* form. The best of all of them is **bold** form. See App. B.3

ited observations. Together with the perplexity improvements, these results reaffirm our assumption that the compression layer gives a subtle but strong signal that improves generative performance.

C Ablation Study

C.1 Impact of parameter sharing of encoder and decoder

To find out the importance of multiple encoders and decoders on two layers, we have used shared parameters on both of them and see the effect on the result. The result for this experiment ($ours_{encdec}$) is reported on Table 14. We can see a substantial drop in the result, especially on the INC score for low perplexity scores (0.5, 0.4, 0.2).

Model	ϵ	Wikipedia INC (Test Data) Base	Compression	Total
HAQAE	-	-	-	24.88 ± 1.35
SSDVAE	0.9	-	-	35.56 ± 1.70
scenario-only		*42.29 ± 1.79*	25.38 ± 1.84	39.86 ± 1.82
SSDVAE	0.7	-	-	*39.08 ± 1.55*
scenario-only		38.79 ± 4.11	26.83 ± 7.32	32.91 ± 7.29
SSDVAE	0.5	-	-	*40.18 ± 0.90*
scenario-only		37.59 ± 5.61	22.06 ± 1.01	35.59 ± 4.71
SSDVAE	0.4	-	-	*47.88 ± 3.59*
scenario-only		40.91 ± 2.19	22.15 ± 1.37	37.99 ± 1.86
SSDVAE	0.2	-	-	44.38 ± 2.10
scenario-only		***48.1 ± 2.22***	20.54 ± 0.1	43.3 ± 2.33

Table 11: Wikipedia Inverse Narrative Cloze Score for test data (higher is better). For each observation probability (ϵ), the best score is in *italic* form. The best of all of them is **bold** form. See App. B.3

Model	ϵ	Perplexity (Test Data) Base	Compression	Total
HAQAE	-	-	-	21.38 ± 0.25
SSDVAE	0.9	-	-	19.84 ± 0.52
grouping		***19.44 ± 0.5***	31.36 ± 0.85	24.69 ± 0.64
SSDVAE	0.7	-	-	21.19 ± 0.76
grouping		*20.13 ± 1.45*	29.7 ± 0.51	24.43 ± 0.84
SSDVAE	0.5	-	-	31.11 ± 0.85
grouping		*21.52 ± 0.72*	31.62 ± 0.51	26.08 ± 0.39
SSDVAE	0.4	-	-	33.12 ± 0.54
grouping		*23.42 ± 0.59*	30.16 ± 4.2	27.45 ± 0.66
SSDVAE	0.2	-	-	33.31 ± 0.63
grouping		*28.17 ± 2.26*	34.17 ± 0.98	31 ± 1.31

Table 12: Per-word perplexity for test data (lower is better). For each observation probability (ϵ), the best perplexity is in *italic* form. The best of all of them is **bold** form. See App. B.4

C.2 Impact of parameter sharing of frame embedding

To determine the importance of multiple frame embedding weights for each layer, we have used one shared frame embedding layer across both layers. We compute results across three seeds. The result for this experiment ($ours_{frame}$) is reported on Table 14. Similar to the encoder-decoder, we can see a substantial decrease in the INC score.

C.3 Impact of summation or concatenation of both layer encoding

To illustrate if both layer encodings altogether can improve the result, we have done two experiments, one with the summation of both layers encodings ($ours_{sum}$) and another with only concatenation of both layer encodings ($ours_{cat}$). Both experiments' results are reported on Table 14. Both of the models have a large drop on INC, which demonstrates the importance of the performance of the individual encoding.

Model	ϵ	Wikipedia INC (Test Data)		
		Base	Compression	Total
HAQAE	-	-	-	24.88 ± 1.35
SSDVAE	0.9	-	-	35.56 ± 1.70
grouping		*40.76 ± 2.86*	28.23 ± 1.04	39.4 ± 1.59
SSDVAE	0.7	-	-	*39.08 ± 1.55*
grouping		38.09 ± 5.6	26.55 ± 0.51	37.83 ± 5.08
SSDVAE	0.5	-	-	*40.18 ± 0.90*
grouping		39.5 ± 3.45	25.61 ± 0.96	37.86 ± 2.56
SSDVAE	0.4	-	-	*47.88 ± 3.59*
grouping		43.83 ± 1.75	24.79 ± 0.43	42.16 ± 1.43
SSDVAE	0.2	-	-	44.38 ± 2.10
grouping		***48.88 ± 1.37***	26.64 ± 0.98	46.81 ± 1.67

Table 13: Wikipedia Inverse Narrative Cloze Score for test data (higher is better). For each observation probability (ϵ), the best score is in *italic* form. The best of all of them is **bold** form. See App. B.4

Model	ϵ	Perplexity (Test Data)			Wikipedia INC (Test Data)		
		Base	Compression	Total	Base	Compression	Total
HAQAE	-	-	-	21.39 ± 0.25	-	-	24.88 ± 1.35
SSDVAE		-	-	19.84 ± 0.52	-	-	35.56 ± 1.70
$ours_{encdec}$		26.25 ± 0.12	26.59 ± 0.13	26.42 ± 0.12	38.35 ± 1.66	38.42 ± 1.53	38.28 ± 1.65
$ours_{frame}$	0.9	20.94 ± 0.86	37.01 ± 1.55	27.83 ± 0.85	41.82 ± 2.44	28.37 ± 4.09	39.97 ± 1.16
$ours_{sum}$		*18.63 ± 0.24*	32.02 ± 4.46	24.38 ± 1.59	40.88 ± 0.25	36.15 ± 11.71	42.65 ± 5.02
$ours_{cat}$		19.34 ± 1.04	31.25 ± 2.07	24.54 ± 0.23	*44.05 ± 0.61*	25.43 ± 4.73	37.53 ± 4.54
SSDVAE		-	-	21.19 ± 0.76	-	-	39.08 ± 1.55
$ours_{encdec}$		27.15 ± 0.64	27.61 ± 0.64	27.38 ± 0.64	40.68 ± 1.78	40.37 ± 1.27	40.52 ± 1.43
$ours_{frame}$	0.7	20.77 ± 0.2	38.75 ± 1.18	28.37 ± 0.33	41.38 ± 3.48	33.22 ± 4.4	41.71 ± 2.75
$ours_{sum}$		*19.51 ± 0.5*	30.37 ± 3.29	24.33 ± 1.61	41.68 ± 1.25	31.77 ± 10.34	40.92 ± 5.04
$ours_{cat}$		20.17 ± 0.42	30.04 ± 2.89	24.59 ± 1.09	*43.42 ± 1.53*	28.15 ± 5.53	39.63 ± 3.45
SSDVAE		-	-	31.11 ± 0.85	-	-	40.18 ± 0.90
$ours_{encdec}$		26.54 ± 1.68	28.79 ± 1.55	27.65 ± 1.61	37.02 ± 5.75	37.03 ± 5.7	36.9 ± 5.9
$ours_{frame}$	0.5	19.55 ± 0.89	37.84 ± 1.72	27.19 ± 0.98	*45.48 ± 3.63*	27.9 ± 1.68	40.7 ± 3.55
$ours_{sum}$		*19.15 ± 0.38*	30.58 ± 1.28	24.19 ± 0.57	41.03 ± 1.32	43.37 ± 2.03	*46.83 ± 1.55*
$ours_{cat}$		19.59 ± 0.22	30.39 ± 1.49	24.4 ± 0.6	41.45 ± 2.05	26.12 ± 4	38.57 ± 5.59
SSDVAE		-	-	33.12 ± 0.54	-	-	***47.88 ± 3.59***
$ours_{encdec}$		25.56 ± 0.53	28.03 ± 0.47	26.77 ± 0.5	36.52 ± 3.06	36.23 ± 2.85	36.57 ± 2.97
$ours_{frame}$	0.4	19.6 ± 1.16	38.03 ± 0.74	27.29 ± 0.58	38.13 ± 2.55	26.78 ± 3.21	37.18 ± 0.73
$ours_{sum}$		18.79 ± 0.98	32.09 ± 1.27	24.56 ± 1.04	43.33 ± 0.88	37.47 ± 14.06	45.82 ± 4.8
$ours_{cat}$		*18.74 ± 0.83*	32.1 ± 2.14	24.52 ± 1.14	42.28 ± 3.73	32.37 ± 9.64	43.2 ± 2.66
SSDVAE		-	-	33.31 ± 0.63	-	-	*44.38 ± 2.10*
$ours_{encdec}$		25.62 ± 0.31	30.85 ± 0.17	28.12 ± 0.1	38.1 ± 3.1	38.32 ± 3.37	38.27 ± 3.22
$ours_{frame}$	0.2	18.63 ± 0.75	38.68 ± 0.36	26.84 ± 0.65	41.45 ± 2.33	29.62 ± 0.98	40.43 ± 2.95
$ours_{sum}$		***17.1 ± 0.21***	29.19 ± 3.06	22.33 ± 1.31	39.25 ± 4.42	31.65 ± 11.88	40.45 ± 7.24
$ours_{cat}$		17.21 ± 0.65	29.6 ± 2.78	22.56 ± 1.18	38.55 ± 0.41	20.45 ± 9.19	29.77 ± 9.34

Table 14: Wikipedia Inverse Narrative Cloze Score for test data (higher is better). For each observation probability (ϵ), the best score is in *italic* form. The best of all of them is **bold** form. See App. C.1, App. C.2, App. C.3