Technical Summary: Video Event Extraction via Tracking Visual States of Arguments

Kai-Yu Lu

1 Research Problem and Motivation

Video event extraction aims to recover event semantic structures from short video clips. For each clip, the target output is an *event frame* that specifies a salient verb and a set of arguments together with their semantic roles, such as `KNOCK(AGENT = bull, TARGET = man, SCENE = ground)`.

Most existing models for image and video situation recognition encode only global frame level visual features and then predict the verb and roles from these holistic representations. Such designs lack explicit modeling of argument level visual dynamics and interactions, which are often decisive for identifying events.

From the linguistic perspective, many events are resultative and telic phenomena that correspond to changes of state. Motivated by this view, the paper formulates video events as the aggregation of visual state changes of all arguments involved in the event. The core hypothesis is that tracking how each argument and the interaction among arguments evolve over time provides the most informative evidence for video event extraction.

This formulation exposes the following main gaps in prior work.

- Global encoders capture overall scene semantics but do not explicitly trace how individual objects move and deform across frames. As a result, events with similar global appearance but different argument level dynamics, such as watching versus knocking down, are difficult to distinguish.
- Previous methods seldom treat the multi object interaction region as a first class representation. Relative motion and interaction pixels between arguments provide strong cues for events such as colliding, fighting, or handing over, yet they are not explicitly encoded.

The proposed framework addresses these gaps by decomposing visual state change into three components:

1. pixel level changes inside object bounding boxes;
2. geometric displacement of bounding boxes;
3. multi argument interaction regions that capture joint appearance and geometry.

2 Related Work

2.1 Image and Video Situation Recognition

Image situation recognition studies visual semantic role labeling in static images. An image is annotated with a verb and a set of semantic roles, and each role is filled with a noun phrase. Models typically use

convolutional neural networks to encode the image and either predict roles directly from global features or incorporate object detectors to refine argument localization.

Video situation recognition extends this formulation to short clips. The VidSitu benchmark defines an ontology of verb senses and argument roles, and provides densely annotated video clips. Baseline architectures in this setting employ 3D convolutional backbones such as I3D and SlowFast, and treat the entire clip as a single feature vector for verb classification and argument prediction.

2.2 Video Action Recognition and Transformer based Models

Action recognition traditionally relies on 3D convolutional neural networks that operate on consecutive frames, for example I3D and SlowFast. More recently, transformer based models such as TimeSformer have been proposed to encode videos with self attention layers applied over spatio temporal tokens. These models achieve strong performance on single label action classification tasks, but they do not explicitly model event structure with multiple arguments and roles.

2.3 Event Extraction and Structured Prediction

Text based event extraction emphasizes that events are structured objects and that arguments and their roles must be modeled jointly. Graph based or structured prediction methods are often used to couple trigger and argument predictions. Multimedia event extraction combines visual and textual cues but has not explicitly focused on argument state trajectories in videos. The present paper can be viewed as importing this structured perspective into video understanding while building on modern 3D encoders and object tracking.

3 Dataset Construction

3.1 VidSitu Dataset

All experiments are conducted on VidSitu, a public video situation recognition dataset that provides extensive verb and argument structure annotations for more than one hundred and thirty thousand short clips. Each video clip is approximately two seconds in length. Clips in the evaluation splits are annotated with ten verbs, while clips in the training split are annotated with a single salient verb. The event ontology contains two thousand one hundred and fifty four verb senses, and each verb has at least three semantic roles.

The dataset statistics are summarized in Table 1. Splits are defined for training, validation, verb evaluation, and role evaluation.

Split	Train	Valid	Test Verb	Test Role
Number of clips	118,130	6,630	6,765	7,990
Number of verb instances	118,130	66,300	67,650	79,900
Number of role instances	118,130	19,890	20,295	23,970

Table 1: VidSitu dataset statistics.

3.2 Video Preprocessing and Object Tracking

Each clip is processed by a SlowFast backbone, which uses:

- a slow pathway that samples frames at a low rate and focuses on spatial semantics;

- a fast pathway that samples frames at a higher rate to capture motion at fine temporal resolution.

An external video relation detection and tracking model, VidVRD, is applied to each clip to generate object tracklets. Each tracklet corresponds to a detected object and contains its bounding boxes and time stamps across the clip. These tracklets serve as candidates for arguments in event structures and provide the basis for state tracking.

4 Query Protocol and Task Definitions

4.1 Event Representation

Given a video clip with sampled frames $\{f_i\}_{i=0}^{F}$, the model is required to output a set of events. Each event is represented as

$$e = \big\{v,\ \langle r_0, a_0\rangle,\ \langle r_1, a_1\rangle,\ \dots\big\}, \tag{1}$$

where:

- v is the verb sense chosen from a finite verb set V defined by the VidSitu ontology;
- $r_k \in R(v)$ is the kth semantic role associated with verb v, where $R(v)$ denotes the role set for this verb, for example $\{\text{AGENT}, \text{TARGET}, \text{SCENE}, \dots\}$ for verb `KNOCK`;
- a_k is the argument entity for role r_k, expressed as a free form word or phrase such as "gray bull" or "ground".

Equation (1) formalizes an event as a verb together with an unordered set of role–argument pairs. This structure is the target prediction for both tasks.

4.2 Task 1: Verb Classification

Verb classification is a motion related video understanding task that selects one or more verbs from the predefined verb ontology. For each clip in the validation or test splits, ten verbs are annotated. The model predicts a ranked list of candidate verbs.

Let $\hat{V}_K$ denote the set of top K predicted verbs for a clip, and let V^* denote the set of annotated ground truth verbs. The following ranking based metrics are used.

- **Accuracy@1**: fraction of clips where the top one prediction lies in V^*.
- **Accuracy@5**: fraction of clips where at least one verb in $\hat{V}_5$ lies in V^*.
- **Recall@5**: fraction of ground truth verbs in V^* that are covered by $\hat{V}_5$.
- **F1@5**: F1 score between the predicted set $\hat{V}_5$ and the ground truth set V^*, combining precision and recall over the top five predictions.

4.3 Task 2: Semantic Role Prediction

Semantic role prediction aims to generate the argument structure for each clip. The output is a sequence that concatenates the verb, explicit role markers, and argument phrases in order, for example:

```
KNOCK [Arg0] gray bull [Arg1] midget in gray hoodie [ArgScene] ground.
```

Arguments are free form noun phrases, therefore caption style generation metrics are adopted.

- **CIDEr**: consensus based evaluation score that measures similarity between generated phrases and multiple reference phrases, emphasizing content words that are important across references.
- **CIDEr Verb**: macro averaged CIDEr over verb senses, by grouping samples according to the verb and averaging CIDEr scores within each group.
- **CIDEr Arg**: macro averaged CIDEr over role types, by grouping samples according to role type and averaging CIDEr scores for each role.
- **ROUGE L**: longest common subsequence based metric that captures the overlap of contiguous subsequences between generated and reference phrases, reflecting phrase level completeness.

These metrics collectively evaluate the quality of generated argument phrases and the stability of performance across diverse verbs and roles.

5 Modeling Approach

5.1 Video Encoder and Object Tracklets

5.1.1 Object Bounding Boxes

An object tracking model is applied to obtain a set of objects $\{o_j\}_{j=0}^{O}$ in the clip. For object o_j in frame f_i, the tracker outputs an appearance bounding box

$$b_{ji} = \big\langle (b_{ji}^1, b_{ji}^2),\ (b_{ji}^3, b_{ji}^4) \big\rangle, \tag{2}$$

where:

- (b_{ji}^1, b_{ji}^2) denotes the coordinate of the lower left corner of the box in the original frame coordinate system;
- (b_{ji}^3, b_{ji}^4) denotes the coordinate of the upper right corner.

These coordinates define the spatial extent of object o_j in frame f_i and are later normalized into the feature grid coordinate system.

5.1.2 SlowFast Video Backbone

The video encoder uses the SlowFast network to map raw frames into spatio temporal feature grids. The two pathways are summarized by

$$\big[g^{\text{slow}},\ g^{\text{fast}}\big] = \text{SlowFast}\big(f^{\text{slow}}, f^{\text{fast}}\big), \tag{3}$$

where:

- $f^{\text{slow}} \in \mathbb{R}^{F_1 \times W \times H \times 3}$ is the slow pathway input, consisting of F_1 frames sampled at a low frame rate;
- $f^{\text{fast}} \in \mathbb{R}^{F_2 \times W \times H \times 3}$ is the fast pathway input, consisting of F_2 frames sampled at a higher frame rate with $F_2 > F_1$;
- $g^{\text{slow}} \in \mathbb{R}^{F_1 \times W' \times H' \times d_1}$ and $g^{\text{fast}} \in \mathbb{R}^{F_2 \times W' \times H' \times d_2}$ are the corresponding grid feature maps, where W' and H' are spatial grid sizes, and d_1 and d_2 are channel dimensions.

These grid features provide high level local descriptors that are used to construct object and interaction state embeddings.

5.2 Object State Embedding

5.2.1 Single Frame Visual State

For an object o_j in frame f_i, the appearance bounding box b_{ji} is first projected into the grid feature space, producing a normalized box

$$\hat{b}_{ji} = \big\langle (\hat{b}^1_{ji}, \hat{b}^2_{ji}),\ (\hat{b}^3_{ji}, \hat{b}^4_{ji}) \big\rangle,$$

where the coordinates are scaled to align with the grid indices of g^{slow} or g^{fast}.

The visual state of object o_j in frame f_i is obtained by average pooling grid features inside this projected bounding box:

$$p_{ji} = \frac{1}{(\hat{b}^3_{ji} - \hat{b}^1_{ji})(\hat{b}^4_{ji} - \hat{b}^2_{ji})} \sum_{x=\hat{b}^1_{ji}}^{\hat{b}^3_{ji}-1} \sum_{y=\hat{b}^2_{ji}}^{\hat{b}^4_{ji}-1} g_i[x, y]. \tag{4}$$

Here:

- g_i denotes the grid feature map of frame f_i from either the slow or the fast pathway;
- $g_i[x, y]$ denotes the feature vector at grid coordinate (x, y);
- the denominator is the area of the projected bounding box in the grid coordinate space.

Vector p_{ji} therefore summarizes the local appearance and motion pattern of the object in the current frame, capturing posture, orientation, and subtle pixel level changes.

5.2.2 Coordinate Embedding and State Vector

To track displacement, the model encodes the coordinates of the bounding box through a positional embedding:

$$c_{ji} = W^c \hat{b}_{ji}, \tag{5}$$

where:

- $W^c \in \mathbb{R}^{d_c \times 4}$ is a learnable linear projection matrix;
- $\hat{b}_{ji} \in \mathbb{R}^4$ concatenates the projected coordinates $(\hat{b}^1_{ji}, \hat{b}^2_{ji}, \hat{b}^3_{ji}, \hat{b}^4_{ji})$;
- $c_{ji} \in \mathbb{R}^{d_c}$ is the coordinate embedding, with d_c denoting its dimensionality.

The *Object State Embedding* for object o_j in frame f_i is defined as

$$s_{ji} = [\, p_{ji},\ c_{ji} \,], \tag{6}$$

which concatenates pooled visual features and coordinate based features into a single state vector $s_{ji} \in \mathbb{R}^{d+d_c}$. This vector represents the instantaneous visual state of the object, combining appearance and location information.

5.3 Object Motion aware Embedding

To capture how an object evolves across multiple frames, the model aggregates its state embeddings along the temporal dimension. For object o_j, the *Object Motion aware Embedding* is

$$m_j = \text{StateAgg}\big(\{s_{ji}\}_{i=0}^{F}\big), \tag{7}$$

where:

- $\{s_{ji}\}_{i=0}^{F}$ is the sequence of state embeddings for object o_j across all frames in which it appears;
- StateAgg is a temporal aggregation operator that combines these state vectors into a single summary;
- $m_j \in \mathbb{R}^{d+d_c}$ is the resulting motion aware embedding.

In practice, StateAgg is instantiated as either a Long Short Term Memory network (LSTM) or as average pooling, and ablation studies compare these alternatives. This encoding is performed separately for slow and fast pathways, producing $m_j^{\text{slow}} \in \mathbb{R}^{d_1+d_c}$ and $m_j^{\text{fast}} \in \mathbb{R}^{d_2+d_c}$, which are then concatenated.

Vector m_j captures translational motion and local appearance changes, such as moving from left to right or changing from standing to falling.

5.4 Object Interaction Embedding

5.4.1 Union Interaction Region

Beyond individual objects, the model also encodes the joint state of multiple objects. For frame f_i, let O_i denote the set of objects present in this frame. The union bounding box of these objects in the grid feature space is defined as

$$\hat{B}_i = \bigcup_{o_j \in O_i} \hat{b}_{ji}, \tag{8}$$

where the union is taken over the projected bounding boxes of all objects in frame f_i. This union box covers the interaction area containing all arguments.

As with individual objects, visual features inside the interaction region are pooled:

$$p_i' = \frac{1}{(\hat{B}_i^3 - \hat{B}_i^1)(\hat{B}_i^4 - \hat{B}_i^2)} \sum_{x=\hat{B}_i^1}^{\hat{B}_i^3-1} \sum_{y=\hat{B}_i^2}^{\hat{B}_i^4-1} g_i[x, y], \tag{9}$$

where $(\hat{B}_i^1, \hat{B}_i^2)$ and $(\hat{B}_i^3, \hat{B}_i^4)$ are the corners of the union box, and $g_i[x, y]$ is the grid feature at coordinate (x, y).

The coordinates of the union box are encoded with the same projection:

$$c_i' = W^c \hat{B}_i, \tag{10}$$

yielding a coordinate embedding $c_i' \in \mathbb{R}^{d_c}$ that captures the location and size of the interaction region.

The interaction state in frame f_i is then

$$\iota_i = [\, p_i',\ c_i' \,]. \tag{11}$$

5.4.2 Interaction Trajectory

To track how interactions evolve across frames, interaction states are aggregated:

$$\iota = \text{InterAgg}\big(\{\iota_i\}_{i=0}^{F}\big), \tag{12}$$

where:

- $\{\iota_i\}_{i=0}^{F}$ is the sequence of interaction state vectors across frames;
- InterAgg is a temporal aggregation operator with the same architecture as StateAgg;
- ι is the *Object Interaction Embedding*.

Embedding ι captures evolving relative positions and joint appearance of multiple objects, for example two objects moving towards each other, overlapping, and then moving apart, together with contextual background changes such as the ground and surrounding structures.

5.5 Argument Interaction Encoder

5.5.1 Input to Transformer Layer

The model aggregates three types of information:

- a global video feature g obtained by average pooling the SlowFast grid features over all frames and spatial locations;
- a set of object motion aware embeddings $\{m_j\}_{j=0}^{O}$;
- the object interaction embedding ι.

These components are concatenated into a sequence and fed into a Transformer encoder layer:

$$e = \text{Trans}\big([\, g;\ m_0;\ m_1;\ \ldots;\ m_O;\ \iota\,]\big), \tag{13}$$

where:

- Trans is a single layer Transformer encoder with self attention;
- e is the resulting *event aware visual embedding*.

The self attention mechanism allows each element (global feature, object embedding, interaction embedding) to attend to all others. This enables the encoder to focus on salient objects and interactions that are most relevant to the event while downweighting irrelevant background objects.

Embeddings from slow and fast pathways are concatenated prior to the Transformer, and a final linear projection maps e into a desired dimensionality for downstream tasks.

5.6 Verb Classification Head

For verb prediction, the event aware embedding e is passed through a two layer feedforward network with ReLU activation, followed by a Softmax function:

$$v = \text{Softmax}\big(W^{(2)}\big[\text{ReLU}\big(W^{(1)}e + b^{(1)}\big)\big] + b^{(2)}\big), \tag{14}$$

where:

- $W^{(1)} \in \mathbb{R}^{h \times d}$ and $b^{(1)} \in \mathbb{R}^{h}$ define a bottleneck layer, with d the dimension of e and h set to half of $d_1 + d_2$;
- $W^{(2)} \in \mathbb{R}^{|V| \times h}$ and $b^{(2)} \in \mathbb{R}^{|V|}$ map the bottleneck representation to logits over the verb vocabulary V;
- Softmax converts logits into a probability distribution over candidate verbs.

The verb classifier is trained with a cross entropy loss between the predicted distribution and ground truth verb labels. Minimizing this loss encourages high probability for annotated verbs and low probability for non occurring verbs.

5.7 Semantic Role Prediction Decoder

Semantic role prediction is formulated as conditional sequence generation. The decoded sequence is the concatenation of the verb and its arguments in the pattern

```
v [Arg0] a_0 [Arg1] a_1 ...,
```

where each token is produced by a Transformer decoder that conditions on previously generated tokens and the visual embedding e. The kth argument a_k is predicted by

$$a_k = \text{Decoder}\big(v,\ [\text{Arg0}],\ a_0,\ [\text{Arg1}],\ a_1,\ \ldots,\ [\text{Arg}k]\ ;\ e\big), \tag{15}$$

where:

- v is the decoded verb token;
- $[\text{Arg}k]$ is a special marker token indicating the start of the kth role;
- Decoder is a Transformer decoder that attends both to the prefix sequence and to the event aware embedding e.

During training, teacher forcing is used. The ground truth previous arguments and the ground truth verb are provided to the decoder, and a cross entropy loss is computed between the generated sequence and the ground truth sequence. Minimizing this loss drives the model to produce accurate role specific argument phrases that are consistent with the visual content.

5.8 Training Configuration

The main hyperparameters are summarized in Table 2. Verb classification and semantic role prediction are trained separately.

6 Empirical Results

6.1 Verb Classification

Relative to the SlowFast baseline without Kinetics pretraining, the best proposed model improves F1@5 on the validation set from 17.87 to 21.40 and on the test set from 19.20 to 22.47. These gains correspond to approximately 3.5 absolute points and substantiate the effectiveness of argument state tracking for verb identification. Accuracy@5 and Recall@5 also improve substantially, indicating better coverage of all ground truth verbs among the top ranked predictions.

Component	Hyperparameter	Value
Verb classification	Number of epochs	10
	Batch size	8
	Learning rate	chosen from $\{10^{-4}, 3 \cdot 10^{-5}\}$
	Optimizer	Adam, $\beta_1 = 0.9$, $\beta_2 = 0.99$, $\epsilon = 10^{-8}$
	Maximum objects per clip	8
	Coordinate embedding dimension d_c	128
	Pretrained backbone	SlowFast with Kinetics 400 pretraining
	Learning rate for pretrained parameters	shrunk by ninety per cent
Semantic role prediction	Number of epochs	10
	Batch size	8
	Learning rate	10^{-4}
	Optimizer	Adam, same settings as above
	Visual encoder	event aware embedding e frozen
	Sequence model	Transformer based encoder and decoder

Table 2: Key training hyperparameters for verb classification and semantic role prediction.

6.2 Semantic Role Prediction

Table 3 reports semantic role prediction performance on the validation split. Results are averaged over ten runs and the standard deviation is reported to reflect stability.

Model	CIDEr	CIDEr Verb	CIDEr Arg	ROUGE L
SlowFast baseline (rerun)	44.49 ± 2.30	51.73 ± 2.70	40.93 ± 2.42	40.83 ± 1.27
Ours (OSE pixel + OME)	47.82 ± 2.12	54.51 ± 3.00	44.32 ± 2.45	40.91 ± 1.32
Ours (OSE pixel/disp + OME)	48.46 ± 1.84	56.04 ± 2.12	44.60 ± 2.33	41.89 ± 1.12
Ours (OSE pixel/disp + OME + OIE)	47.16 ± 1.71	53.96 ± 1.32	42.78 ± 2.74	40.86 ± 2.54

Table 3: Semantic role prediction results on VidSitu validation split. Values are averages over ten runs with standard deviation.

The best configuration, which uses Object State Embedding with both pixel and displacement components together with Object Motion aware Embedding, improves CIDEr from 44.49 to 48.46. CIDEr Verb and CIDEr Arg also increase, indicating improved performance across verbs and roles. Additionally, the standard deviation decreases from 2.30 to 1.84 for CIDEr, which suggests that fine grained argument state tracking leads to more stable training and more robust predictions.

6.3 Ablation Studies

6.3.1 Pixel Changes versus Displacements

Comparison between OSE pixel and OSE pixel/disp variants demonstrates the contribution of displacement features. Adding coordinate displacement information consistently improves verb classification and semantic role prediction. This confirms that modeling object trajectories explicitly, in addition to local pixel changes, is beneficial for characterizing motion.

6.3.2 Effect of Object Interaction Embedding

Adding Object Interaction Embedding improves verb classification slightly, as verbs usually describe interactions among entities. However, for semantic role prediction, Object Interaction Embedding does not yield further gains and sometimes slightly reduces performance. This indicates that interaction features are particularly helpful for deciding the verb, while precise argument naming primarily relies on each individual object's motion and appearance.

6.3.3 Number of Objects

The maximum number of objects per clip is varied among two, four, and eight. Results for verb classification on the validation split are shown in Table 4.

Setting	Acc@1	Acc@5	Rec@5	F1@5
$O_{\max} = 2$	53.56	83.98	28.95	21.53
$O_{\max} = 4$	53.56	84.10	28.51	21.29
$O_{\max} = 8$	53.35	83.94	28.72	21.40

Table 4: Effect of the maximum number of objects on verb classification performance.

The differences across these settings are minor, which suggests that the Argument Interaction Encoder can automatically focus on informative objects and downweight less relevant ones, such as background items.

6.3.4 State Aggregator Choice in OME

Table 5 compares different choices for the state aggregation operator in Object Motion aware Embedding.

Aggregator	Acc@1	Acc@5	Rec@5	F1@5
LSTM	53.23	83.76	28.57	21.30
Average pooling	53.32	83.97	28.64	21.36

Table 5: Effect of different state aggregation operators in Object Motion aware Embedding.

The performance difference between LSTM and average pooling is small. Average pooling is slightly more efficient and achieves marginally higher scores, indicating that a few parameters are sufficient to encode object state changes over time.

6.4 Qualitative Analysis

Qualitative examples illustrate how the model exploits argument state changes. In one case, a person's lips are moving, but there is no tracked food object interacting with the person. The proposed model predicts events corresponding to TALK or SPEAK, while a baseline model predicts CHEW because it only observes mouth motion. This example highlights the importance of checking interactions between relevant arguments rather than relying solely on local motion.

In another case, a person moves downward while displaying agitated arm motions. Baseline models tend to predict ROLL or FALL based on gross positional changes, whereas the proposed model predicts FIGHT or HIT by leveraging pixel level changes of the arms and the interaction region between persons. These

examples support the claim that explicit modeling of object state and interaction trajectories leads to more discriminative event representations.

7 Summary

7.1 Research Problem and Motivation

The study addresses the problem of video event extraction, which requires predicting both verbs and structured argument roles for short clips. The motivation is that existing video situation recognition methods rely primarily on global frame level features and do not explicitly model argument level visual state changes or interactions. By formulating events as aggregations of argument state changes, the work aligns video representations with linguistic views of events as changes of state.

7.2 Key Modeling Contributions

The main modeling contributions are as follows.

- Introduction of an argument centric formulation where events are represented as verbs together with argument state trajectories, rather than purely global scene descriptors.
- Decomposition of visual state change into in bounding box pixel dynamics, bounding box displacements, and multi object interaction regions, each encoded by dedicated embeddings.
- Design of an Argument Interaction Encoder that integrates global, object level, and interaction level information through a Transformer layer, enabling the model to attend to salient arguments and interactions.
- Integration of the event aware visual embedding into both verb classification and semantic role prediction, replacing the standard SlowFast features used in prior work.

7.3 Empirical Findings

Experimental results on the VidSitu benchmark show the following.

- For verb classification, the proposed model achieves approximately three and one third absolute points of improvement in F1@5 on the hidden test split compared with a strong SlowFast baseline, with similar gains on Accuracy@5 and Recall@5.
- For semantic role prediction, the best configuration improves CIDEr by about four absolute points and reduces the variance across runs, demonstrating both higher accuracy and greater robustness.
- Ablation studies confirm that bounding box displacements and interaction regions contribute meaningfully to performance, while the number of objects and the choice of state aggregator have relatively minor influence.

7.4 Limitations and Future Directions

Several limitations remain.

- The framework depends on the quality of external object detection and tracking. Missed objects or fragmented tracklets propagate errors into state embeddings and event predictions.

- The overall pipeline is computationally intensive, combining a 3D convolutional backbone, object tracking, multiple aggregation modules, and Transformer based encoders and decoders. This may hinder application in resource constrained scenarios.
- Experiments are limited to the VidSitu dataset and short clips. Generalization to longer videos and other domains, as well as to different event ontologies, is not demonstrated.
- The verb and role inventory is fixed by the dataset; the model does not directly support open vocabulary events without further adaptation.

Future work can explore joint training of detection, tracking, and event extraction so that object state trajectories and event predictions are optimized together. Pretraining strategies that expose the model to diverse event patterns before fine tuning on VidSitu, and applications of the event aware embedding to tasks such as video captioning, temporal grounding, and multimodal event extraction, are also promising directions.