Multi-Relational Script Learning for Discourse Relations

I-Ta Lee and **Dan Goldwasser**
Department of Computer Science, Purdue University
{lee2226, dgoldwas}@purdue.edu

Abstract

Modeling script knowledge can be useful for a wide range of NLP tasks. Current statistical script learning approaches embed the events, such that their relationships are indicated by their similarity in the embedding. While intuitive, these approaches fall short of representing nuanced relations, needed for downstream tasks. In this paper, we suggest to view learning event embedding as a multi-relational problem, which allows us to capture different aspects of event pairs. We model a rich set of event relations, such as *Cause* and *Contrast*, derived from the Penn Discourse Tree Bank. We evaluate our model on three types of tasks, the popular Mutli-Choice Narrative Cloze and its variants, several multi-relational prediction tasks, and a related downstream task—implicit discourse sense classification.

1 Introduction

Representing world knowledge that can be used for commonsense reasoning is a long-standing AI goal. *Scripts* (Schank and Abelson, 1977) are structured knowledge representations capturing the relationships between prototypical event sequences and their participants in a given scenario. For example, given the event "*John shot Jim with a gun*", we can infer that "*he got arrested by police*" is more probable than "*he fell asleep*".

In recent years, the problem of extracting script knowledge from text has attracted significant attention. Early works (Chambers and Jurafsky, 2008) focused on symbolic event representations and used Pointwise Mutual Information (PMI) between events to capture their relationships. Recent works (Pichotta and Mooney, 2016a; Granroth-Wilding and Clark, 2016; Wang et al., 2017; Lee and Goldwasser, 2018; Li et al., 2018) represent events using dense vectors, based on event co-occurrence, and use vector similarity over their embeddings to measure their relationship.

Our main observation in this paper is that while models for learning script knowledge improved significantly over the last decade, these models can essentially represent only a single event relationship, co-occurrence. That is, events appearing in similar contexts tend to have similar representations. Although this idea works well for a lot of NLP tasks, it is too coarse for modeling commonsense, which should account for fine-grained relationships. To better understand this, consider the example described in Figure 1. Given the first event, corresponding to the sentence "*Jenny went to her favorite restaurant.*", called Step 1, any of the following events in Step 2 would be highly related, and thus similar, to the input event. That is, "*It was raining outside*" and "*She was very hungry*" are both possible NEXT events. Using event similarity alone is too coarse to support many relevant inferences. However, if the relation between the events is given, more clues can be applied to support reliable inferences. In Figure 1, given Step 2 is a *Reason* to Step 1, analogous to asking the question "*Why did Jenny go there?*", the event "*She was very hungry*" is clearly a more reasonable choice. Therefore, using event similarity alone is too coarse to support many relevant inferences, i.e., capturing the *Reason* for the event, should produce a different set of relevant events, compared to *Temporal* (next) events

To help prioritize between showing diverse types of event relations and providing a framework for this discussion, we focus on a set of discourse relations, introduced by Penn Discourse Tree Bank (PDTB) (Prasad et al., 2007). Traditional script learning models would fall short of making the inferences here. For example, the last inference step in Figure 1 asks for an event that *Contrasts* with the previous step. Based on human commonsense, we can identify that the most probable scenario is "She ordered a meal **but** she liked the food better last time." Modeling the re-

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4214–4226
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

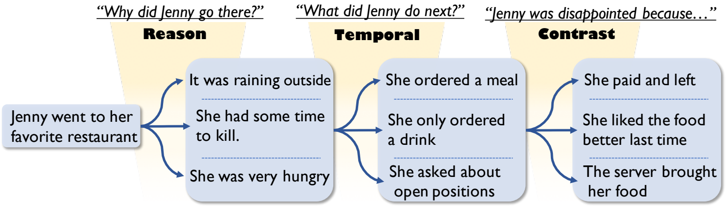

"Why did Jenny go there?"
Reason
"What did Jenny do next?"
Temporal
"Jenny was disappointed because…"
Contrast
Jenny went to her favorite restaurant
It was raining outside
She had some time to kill.
She was very hungry
She ordered a meal
She only ordered a drink
She asked about open positions
She paid and left
She liked the food better last time
The server brought her food

Figure 1: Multi-relational commonsense inference requires different relation types, beyond similarity.

lation type helps us capture different expectations about subsequent events. We use the fact that these relations are often indicated by discourse markers (e.g., "but", capturing the contrasting relation) to extract supervision for learning these relations.

Our goal in this paper is to support such inferences. We introduce a multi-relational event embedding approach, which generalizes the notion of event embedding, by allowing it to capture multiple fine-grained relationships. Our approach builds on recent translation-based embeddings (Bordes et al., 2013; Lin et al., 2015), originally introduced in the context of knowledge graph completion. We adapt these methods to the textual inputs, and suggest a compositional neural network used for capturing the event's internal linguistic structure, while using the translation-based embedding objective to capture different relationships between events. We include 11 relation types, capturing the progression of the narrative: COREF_NEXT, the next event in the coreference chain; NEXT, the next event that occurs subsequently in text; and 9 discourse relations, collectively refer to as DISCOURSE_NEXT.

We evaluate our model in three settings. In the first, we evaluate it on a common benchmark, Multiple-Choice Narrative Cloze (MCNC) task (Granroth-Wilding and Clark, 2016), and its sequential variants proposed by Lee and Goldwasser (2018). We show that we can outperform previously published work by a large margin. In the second setting, we further examine our model's characteristics on three intrinsic tasks. In the last setting, we conduct a challenging downstream task—implicit discourse sense classifications, examplifying the model's applicability.

2 Related Work

Statistical Script Learning was popularized by Chambers and Jurafsky (2008), framing the problem as an unsupervised learning problem, using a PMI-based learning model to approximate a conditional probability of event occurrence. Recent approaches build on representation learning techniques, by learning event embeddings with neural networks. Granroth-Wilding and Clark (2016) utilized Skip-Gram (Mikolov et al., 2013) and an event compositional neural network to adjust event representations. Pichotta and Mooney (2016b; 2016a) applied a LSTM Recurrent Neural Network (RNN), coupled with Beam Search, to model event sequences and their representations. Weber et al. (2017) used three-dimensional tensor-based networks to construct the event representations. Lee and Goldwasser (2018) trained the event embedding with additional features in a hierarchical architecture. Li et al. (2018) constructed an event graph and utilized its network information to make script event predictions. In this paper we combine GRU (Chung et al., 2014), for encoding fine-grained argument information, with a compositional network to generate event representations. GRU was shown to be a competitive alternative to LSTM while requiring less parameters (Kiros et al., 2015; Hochreiter and Schmidhuber, 1997).

Modeling multi-relational data was originally explored for Knowledge Graph Completion, typically focusing on a family of translation-based embedding models which view relations as translations in the vector space. For example, *TransE* (Bordes et al., 2013), captures the relation between h, t, r (embedding of arg0, arg1, and relation), by minimizing the distance between $h + r$ and t. *TransH* (Wang et al., 2014) and *TransR* (Lin et al., 2015) projects the entities into relation-specific spaces. Recent models address issues, such as maintaining structures (Xie et al., 2016; Yoon et al., 2016) and capturing richer interactions (Nickel et al., 2016). In this paper, we adapt *TransE* and *TransR* for narrative script learning, which is an innovative generalization of relation embedding for commonsense inference.

Several recent works looked at modeling specific relationships between events and extracting commonsense knowledge. Zhao et al. (2017) explored modeling cause-effect relations between events; Sap et al. (2018) focused on If-Then relations and showed that their joint multi-task model outperforms the models trained in isolation, based on human evaluations. Peng and Roth (2016) utilized discourse markers to extract relations between semantic frames and modeled them with prevalent language models. Event2Mind (Rashkin et al., 2018) created a dataset capturing the relationship between an event description and its participants' intent and emotional reaction. This idea is related to our work, as the intent and reaction can correspond to *Reason* and *Result* discourse relations in our case. Our goal in this paper is to present a relational generalization over such relationships using a shared embedding space.

The Narrative Cloze (NC) task (Chambers and Jurafsky, 2008) was introduced to evaluate statistical script models by removing an event from a chain, and observing the ranking of the correct answer over the entire event vocabulary, given the rest of the chain. However when complex event structures were considered, e.g., multi-argument events (Pichotta and Mooney, 2014), the large vocabulary size introduced both computational issues and ambiguity into the evaluation. As a result, Granroth-Wilding and Clark (2016) proposed a multiple-choice variation, called MCNC. It simplifies the evaluation process and reduces its computational burden. A similar choice of the multiple-choice adaptation could also be found in recent works, such as Story Cloze (Mostafazadeh et al., 2017) and SWAG (Zellers et al., 2018). In this paper, we evaluate our models on MCNC, and two recent variants (Lee and Goldwasser, 2018) turning MCNC into a sequential inference task. We also introduce relation-specific evaluation capturing the ability of our model to account for nuanced relations beyond co-occurrence.

3 Model

We propose a learning framework, which accounts for the internal predicate-argument structure of events, tuning it to respect different relation types.

Overview Our framework has two preprocessing phases: Event Extraction and Relational Triplet Extraction. In Event Extraction, we aim to identify events from free-form text. The process builds on a dependency parser and coreference resolution. Once events are extracted, we address their relations, specifically three types: (1) events with coreferent entities, (2) events located near each other, and, more importantly, (3) events connected with discourse relations.

The output of the preprocessing phases is a set of relation triplets (e_h, e_t, r), where e_h and e_t are head and tail events, and r is their relation type. We then feed them to a neural network for learning event and relation embeddings. The network objective is an energy function $f(e_t, e_h, r)$, which can be used to approximate the conditional probabilities $p(e_t|e_h, r)$ or $p(r|e_t, e_h)$. This objective captures commonsense knowledge expressed in event relations and embeds it in a vector space, which can be utilized in downstream tasks. Two model variants are proposed in this paper. The first model, *EventTransE*, assumes that all the relations are in the same embedding space and jointly learns representations for events and relations. It works well in some cases, though it might not be expressive enough in others. The second model, *EventTransR*, addresses this issue by introducing relation-specific parameters, which project events into relation-specific spaces when measuring their relatedness.

3.1 Event Extraction

We construct a preprocessing pipeline to extract events and relations over a large text collection. Each event e consists of three components: predicate $(pred(e))$, subject $(subj(e))$, and object $(obj(e))$. Due to computational considerations we restrict the event representation to two arguments. We use a special empty argument representation, *NONE*, for events that have fewer arguments. To obtain the event representation from text, we first run a dependency parser and coreference resolution [1] to acquire the needed information.

Events are extracted by connecting entity mentions on the coreference chain with their corresponding predicate and additional argument, based on the dependency tree. E.g., given, "*Jenny went into her favorite restaurant,*" we extract *(go_into, jenny, her favorite restaurant)*.

Unlike the previous works (Lee and Goldwasser, 2018; Granroth-Wilding and Clark, 2016; Pichotta and Mooney, 2016a), which only consider headwords of entity mentions, we use complete mention spans. In our running example, we consider the object as "*her favorite restaurant*", rather than just "*restaurant*". This allows the models to capture the nuanced information relevant for

[1] Stanford CoreNLP (Manning et al., 2014).

DISCOURSE_NEXT		
Complete	Abbrev.	#relations.
Comparison.Contrast	**Contrast**	7334K
Contingency.Cause.Reason	**Reason**	2818K
Contingency.Cause.Result	**Result**	228K
Contingency.Condition	**Cond.**	3745K
Expansion.Restatement	**Restat.**	16K
Expansion.Conjunction	**Conj.**	98K
Expansion.Instantiation	**Instan.**	249K
Temporal.Synchrony	**Sync.**	63K
Temporal.Asynchronous	**Async.**	379K

Table 1: DISCOURSE_NEXT relations from PDTB and the number of relations extracted for training.

many commonsense inferences, such as the "favorite" here. Other preprocessing steps follow the previous works and are detailed in the appendix.

3.2 Relational Triplet Extraction

Relations are expressed as triplets (e_h, e_t, r), where r is the relation type, and e_h and e_t are events that have an internal structure of $(pred(e), subj(e), obj(e))$. 11 types of relations are considered in this paper for demonstrations: COREF_NEXT, NEXT, and 9 discourse relations, which collectively refer to as DISCOURSE_NEXT.

COREF_NEXT captures sequential relationships between events on the same coreference chain. The NEXT relation is defined between events pairs that co-occurr in a fixed-sized ($w_{context}$) context window. It aims to capture related events that do not share arguments. For example, in "*The forest was on fire. Trees burned.*", the two events do not share arguments, but they often co-occur, and thus are related. Previous works about script learning (Pichotta and Mooney, 2016a; Granroth-Wilding and Clark, 2016; Wang et al., 2017; Lee and Goldwasser, 2018; Li et al., 2018) use either COREF_NEXT or NEXT independently, which failed to leverage the shared information.

For DISCOURSE_NEXT, 9 discourse relations, taken from PDTB, are denoted in Table 1. These relations correspond to commonsense judgments. For example, we can do causal inference with the *Reason* and *Result*; or we can identify the juxtaposition between events by utilizing *Contrast*.

The discourse relations can be represented with a relation type and a pair of argument spans (Xue et al., 2016). For example, "*Jenny went to a restaurant, because she was hungry*" has a relation *Reason* and the spans are the two clauses (omitting the connective "because"). Since training event embedding requires significantly more data than annotated in the PDTB corpus, we approximate this by building a rule-based annotator. We first identify explicit discourse connectives, such as "because," and assume that the surrounding clauses are their argument spans. To determine the relation type, we map the connectives to their most probable type based on the PDTB data. To mitigate the noise, we only take connectives that are highly indicative of their type (85% of connective occurrences are of that type). Note that in our setup a given pair of events might have up to three relations annotated: a discourse relation, NEXT, and COREF_NEXT. We create negative examples by corrupting the positive triplets, randomly replacing e_h, e_t, or r with an event or relation. For each positive triplet we sample one negative triplet. While our weakly supervised relation extraction is noisy, we demonstrate empirically its ability to capture these relations.

3.3 Compositional Event Representation

Figure 2 shows the architecture of our models. Each event e has a raw representation $(pred(e), subj(e), obj(e))$. The predicate $pred(e)$ is given in an embedding lookup table, a matrix with size $|P| \times d_a$, where P denotes predicate vocabulary. $subj(e)$ and $obj(e)$ are encoded with two separate Bi-GRUs (Chung et al., 2014). We call them subject encoder and object encoder, as shown in the figure. The outputs of the encoders are d_a-dimensional respectively. Each GRU is defined as follows:

$$
\begin{aligned}
z_t &= \sigma(W^{(z)} x_t + U^{(z)} h_{t-1}) \\
r_t &= \sigma(W^{(r)} x_t + U^{(r)} h_{t-1}) \\
\bar{h_t} &= tanh(W x_t + r_t \odot U h_{t-1}) \\
\vec{h_t} &= z_t \odot \vec{h_{t-1}} + (1 - z_t) \odot \bar{h_t},
\end{aligned}
$$

where x_t is the input token at timestamp t; $W^{(z)}, U^{(z)}, W^{(r)}, U^{(r)}, W, U$ are parameters to be trained; $\vec{h_t} \in R^{\frac{d_r}{2}}$ is the hidden memory at timestamp t; z_t and r_t are update and reset gates for controlling purposes. The final argument representation is the concatenation of GRU hidden representations trained in two directions, i.e., $h_t = [\vec{h_t}, \overleftarrow{h_t}]$.

The encoded representations for each event component are then fed into a Event Composition network. The network is fully-connected and has one hidden layer, defined as follows:

$$
\begin{aligned}
h_1 &= relu(W_1 x_e + b_1) \\
e &= W_2 h_1 + b_2,
\end{aligned}
$$

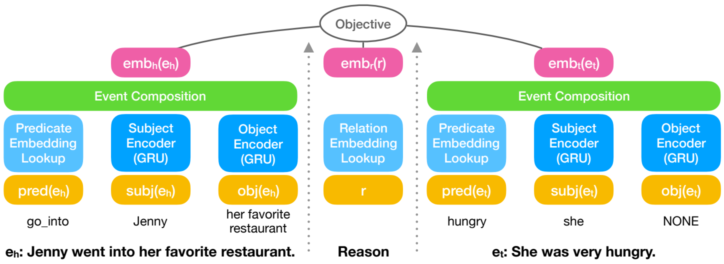


Figure 2: Multi-Relational Script Learning Architecture: the left and right networks encode the event embeddings for e_h and e_t; the middle part encodes the relation r. The training objective on top jointly learn these embeddings.

where x_e is the concatenation of the encoded predicate, subject, and object; $W_1 \in R^{d_h \times 3d_a}, b_1 \in R^{d_h}, W_2 \in R^{d_r \times d_h}, b_2 \in R^{d_r}$ are model parameters. The output $e \in R^{d_r}$ is the event embedding.

For the relations, we embed them using another embedding lookup table. The table size is $n_{rel} \times d_r$, where n_{rel} is the number of relation types. In our case, $n_{rel} = 11$.

3.4 Model: EventTransE

EventTransE is an event embedding model inspired by *TransE* (Bordes et al., 2013). The idea is to embed nodes and their relations in the same vector space so that the distance between nodes reflects their relations. This is called translating operations in the original paper. Based on this idea, we explore a new possibility of learning event embeddings that can make inferences conditioned on different relations. We connect the *TransE* objective to the previous compositional network outputs, which can be formulated as follows:

$$
\begin{aligned}
f_{transe}(t) &= f_{transe}((e_h, e_t, r)) \\
&= \|e_h + r - e_t\|_p^p,
\end{aligned}
\qquad (1)
$$

where $e_h, e_t, r \in R^{d_r}$ are the embeddings from the Event Composition network. Note that Equation 1 is a dissimilarity measure. Lower scores mean that the given two events are strongly related.

3.5 Model: EventTransR

A known issue of *EvenTransE* is its limited ability to deal with reflexive, 1-to-N, N-to-1, or N-to-N relations (Wang et al., 2014). Consider a simple example illustrating the problem: given Equation (1), it is possible to learn a zero relation vector r and two arbitrary but identical event representations e_h and e_t, which minimize the loss. *EventTransR* is proposed to address these issues by separating the event and relation spaces as *TransR* (Lin et al., 2015). It introduces relation-specific parameters to model the interactions between the spaces. *EventTransR* is defined as follows:

$$
\begin{aligned}
f_{transr}(t) &= f_{transr}((e_h, e_t, r)) \\
&= \|e_h M_r + r - e_t M_r\|_p^p,
\end{aligned}
\qquad (2)
$$

where $r \in R^{d_r}$, $e_h, e_t \in R^{d_e}$ are the input embeddings, and $M_r \in R^{d_e \times d_r}$ is the relation-specific parameters introduced.

3.6 Training Objective

The objective is the Margin-Based Ranking Loss:

$$
L(t) = \sum_{t \in T} \sum_{t^* \in T^*} max(0, \delta + f(t) - f(t^*)),
\qquad (3)
$$

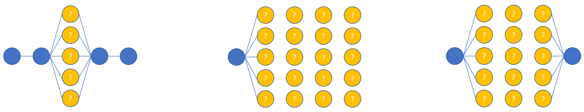
where T is the set of positive relational triplets; T^* is the set of corrupted relational triplets; δ is the margin, and $f \in \{f_{transe}, f_{transr}\}$. At test time, we can leverage the dissimilarity measures to either predict the tail event given the head event and relation, or predict the relation given the head and tail events:

$$
\begin{aligned}
\hat{e_t} &= \arg\min_{e^* \in E} f(e_h, e^*, r); \\
\hat{r} &= \arg\min_{r^* \in R} f(e_h, e_t, r^*).
\end{aligned}
$$

E and R are the event and relation vocabulary.

4 Experiments

We divided our experimental evaluation into three parts. The first focuses on comparing our models with previous work on several common script learning evaluation tasks. The second evaluates our model's ability to capture different relation types between events. In the third, we apply our models to a related downstream task, implicit discourse sense classification, and achieve competitive results by combining our event embeddings



(a) MCNC (b) MCNS (c) MCNE

Figure 3: Comparing single-step prediction (MCNC) and multiple-step inference (MCNS and MCNE).

with ELMo (Peters et al., 2018), a contextualized word embedding model. We provide additional qualitative analysis, showing inferences made by our model, in the appendix.

For training, we use the New York Times (NYT) section of the English Gigaword (Parker et al., 2011). It contains 2M newswire articles and splits into train/dev/test sets, replicating the setup given by Granroth-Wilding and Clark (2016). 500M triplets are extracted from the training set. All the experimental results are averaged over 5 runs. We leave the details about hyperparameter tuning in the appendix. The source code and pre-trained models are publicly available [2].

4.1 Multiple Choice Narrative Cloze Tasks

We begin by evaluating our model on three event representation tasks: Multiple-Choice Narrative Cloze (MCNC), Multiple-Choice Narrative Sequence (MCNS), and Multiple-Choice Narrative Explanation (MCNE). MCNC, proposed by Granroth-Wilding and Clark (2016), measures script learning models' ability to predict a missing event, given its context, in a multiple-choice setting. This evaluation task is not perfect, as noise would be introduced by automatic extraction tools, but not so common as to invalidate the results, and thus this evaluation is widely accepted. Lee and Goldwasser (2018) generalized this single-step task, and suggested two sequence inference versions—MCNS and MCNE. Figure 3 explains the three tasks. Given an event chain, MCNC chooses one step as a multiple-choice question and generates four negative choices for that step. MCNS turns it into a sequence prediction problem by creating multiple-choice questions for each step, except the start event. MCNE provides an additional clue, which is the end event. The inference model has to connect the start and end by explaining things happened in between.

Following the setup in (Granroth-Wilding and Clark, 2016), we evaluate on top of coreferenced event chains, where a protagonist participates each event. The minimum length of the event chains is 9, as short chains are likely to be caused by parsing errors. Our models naturally score the candidates with our training objective $f \in \{f_{trane}, f_{transr}\}$ using COREF_NEXT relation, while other baselines use cosine similarity.

4.1.1 Multiple-Choice Narrative Cloze

We compared two versions of our models, using the entire argument span, or just its headword, with several recently published results.

We compare our models with the following baselines on the MCNC:

- **Random** uniformly selects a candidate.
- **PPMI** (Chambers and Jurafsky, 2008) uses co-occurrence information and calculates Positive PMI for event pairs.
- **BiGram** (Jans et al., 2012) calculates bi-gram conditional probabilities $P(e2|e1)$ based on event term frequencies.
- **Word2Vec** (Mikolov et al., 2013) refers to the pre-trained word embeddings from Word2Vec SkipGram. The summation of word embeddings of predicates and argument mentions are used to represent events.
- **EvSkipGram** (Granroth-Wilding and Clark, 2016) uses SkipGram to learn representations from "sentences" formed by predicates and argument headwords.
- **EventComp** (Granroth-Wilding and Clark, 2016) uses a neural network to learn a compositional function for EvSkipGram and outputs a coherence score for event pairs.
- **SGNN** (Li et al., 2018) is a graph-based model specifically designed for MCNC. It considers each event chain as a sub-graph, and feed it into their GRU-based recurrent networks, which outputs relatedness scores for the candidates.
- **FEEL** (Lee and Goldwasser, 2018) is an event embedding model that does multi-task learning for inter-event relations and intra-event features.
- **PairLSTM** (Wang et al., 2017) is an event embedding model that considers event order information and uses a LSTM network's hidden states for event representations.

Since we need the complete argument spans for events, which is not available in (Granroth-Wilding and Clark, 2016)'s pre-processing proce-

[2] https://github.com/doug919/multi_relational_script_learning

Methods	**Accuracy**
Random*	20.00
PPMI (Chambers and Jurafsky, 2008)	30.52
BiGram (Jans et al., 2012)	29.67
Word2Vec* (Mikolov et al., 2013)	37.39
EvSkipGram* (Granroth-Wilding et al., 2016)	46.28
EventComp (Granroth-Wilding et al., 2016)	49.57
FEEL* (Lee et al., 2018)	51.62
SGNN (Li et al., 2018)	52.45
PairLSTM (Wang et al., 2017)	55.12
EventTransE-headword*	60.50
EventTransR-headword*	59.38
EventTransE*	**63.67**
EventTransR*	62.86

Table 2: Accuracy scores (%) of MCNC. **-headword** stands for using headwords only in argument mentions. The star sign (*) denotes that the results are based on the newly sampled evaluation set.

Methods	**NS-V**	**Base-Inf**	**Sky-Inf**	**NE-V**
GloVe	29.38	27.60	38.50	31.29
FEEL	41.60	38.50	46.00	44.80
EventTransE	**59.48**	**51.22**	**64.47**	**60.94**
EventTransR	58.66	50.73	63.65	60.00

Table 3: Acc (%) of MCNS (NS) and MCNE (NE) tasks. $\{$**NS,NE**$\}$**-V** use Viterbi for inference. **Base-Inf** is a local greedy model using the previous prediction only, and **Skyline-Inf** is given gold contextual events when calculating transition probabilities.

dure, we re-implement the event extraction step by carefully following their procedure. We mark the results based on the newly sampled evaluation set with a star sign (*). We released the newly sampled evaluation set for future comparisons. Table 2 shows the results.

Our models outperform the best baseline model for more than 7% absolute accuracy score. We attribute the improvement to three factors: (1) our models encode complete argument mentions rather than just headwords, *EventTranseE-headword* and *EventTransR-headword*, which are our models' variants that use only headwords for arguments, show that about 3% of the improvement is from this; (2) our models have shared event representations over multiple relations, which regularize the representations in diverse aspects, while other baselines do not make use of relations other than Coref_Next. (3) our models' training objective directly measures relation-specific dissimilarity between events, while most others are based on simple cosine similarity.

4.1.2 Multiple-Choice Narrative Sequence

The MCNC looks at a single transition between events; however, it does not capture the flow of the entire narrative. Lee and Goldwasser (2018) proposed MCNS, which instead of sampling candidate options for one event, it samples options for all the events on the chain, except the first event which is used as the starting point for predictions (Figure 3b). Based on the dissimilarity scores calculated by our models, we can compute transition probabilities for each step. Then we can find the most likely sequence using Viterbi inference algorithm (Viterbi, 1967). We follow the evaluation setting used in (Lee and Goldwasser, 2018) and compare three decision models: (1) *Viterbi*, which finds the most probable sequence of predictions; (2) *Baseline-Inf*, which greedily picks the best transition at each step based on the previous prediction; (3) *Skyline-Inf*, which breaks down a sequence of decisions into local decisions, each using the gold states of all the contextual events.

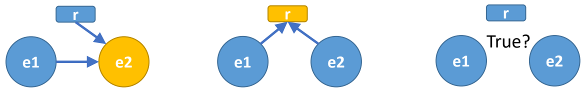
Table 3 shows the results. Our models outperform FEEL (Lee and Goldwasser, 2018), who introduced the task. The same set of reasons given in the section MCNC explain the improvement. We also note that *EventTransE* is especially strong in making predictions for Coref_Next.

4.1.3 Multiple-Choice Narrative Explanation

MCNE is another extension to MCNC. Essentially, in addition to the first event, the final event is also given (Figure 3c). Intuitively, the goal of this evaluation task is to capture explanations, consisting of event sequences, that connect the start and end points. The same inference algorithms as MCNS are adopted. The right three columns of Table 3 gives the result (Note that the *Baseline-Inf* and *Skyline-Inf* are shared with MCNS). The result shows a similar trend as MCNS, but with higher scores, due to the additional information brought by the last event. Note that when calculating the accuracy, we only consider the event blanks in the middle (ignoring the last prediction made in MCNS) for both MCNS and MCNE. This ensures a fair comparison.

4.2 Intrinsic Discourse Relations Evaluation

We suggest three intrinsic tasks, depicted in Figure 4, evaluating how multi-relational information is captured. Given a triplet $(e1, e2, r)$: (1) predict the next event $e2$, (2) predict the relation r, and



(a) (e1, ?, r) (b) (e1, e2, ?) (c) (e1, e2, r)?

Figure 4: Three intrinsic tasks for evaluating our models: (4a) predicts the next event given an event and a relation; (4b) predicts the relation given a pair of events; (4c) binary classification for triplets.

Methods	Accuracy (%)
Random	20.00
ELMo (Peters et al., 2018)	42.97
EventTransE-Random	52.78
EventTransR-Random	26.11
EventTransE-NEXT	51.80
EventTransR-NEXT	51.71
EventTransE	54.83
EventTransR	**55.08**

Table 4: Accuracy scores (%) of the next event prediction, given an event and a relation. **ELMo** is a contextualized word embedding model. **-Random** and **-NEXT** are our model variants that replace the given relation with a random and NEXT relation respectively.

(3) predict its correctness (triplet classification).

Predict the Next Event Similar in spirit to the setup described in Fig. 1, we ask whether knowing the relation, connecting the head to the tail event, would change the expectation about the tail event.

Given a set of triplets that have discourse relations, for each triplet, we corrupt e_t and sample four extra negative choices to form a multiple-choice question. We compare our model variants with a strong baseline model—ELMo (Peters et al., 2018). ELMo is a context-aware word embedding model that has shown strong performance in language understanding tasks. To get the contextualized word embeddings, we have to provide the context, usually the sentence where the target words appear. To retrieve the context, for each event e, we re-construct its "sentence" by concatenating its $subj(e)$, $pred(e)$, and $obj(e)$. The averaged word embedding of the context is used to represent the event. ELMo predicts the next event based on cosine similarity, disregarding the relation. We also make two variants to show our models' awareness to relation types. One replaces the correct discourse relation with a random relation; the other replaces it with a NEXT relation.

Table 4 shows the results. We can see that all our model variants outperform the ELMo baseline, as our models are aware of the relation between events. Similarity-based models that can capture frequently co-occurred events fail to consider the nuanced relations. *EventTransR* performs the best as it has relation-specific parameters emphasizing the relational nuances. Interestingly, using **-NEXT** relation only is also very indicative for predicting the next event, which explains why previous works failed to address the nuanced relations. The results for **-Random** relations indicates that *EventTranR* is very sensitive to incorrect relations. This is due to the separation between the relation and event embedding spaces, useful for relation-sensitive tasks. Also, that **EventTranE-Random** model works better than **EventTranE-NEXT** suggests that our models with discourse relations do capture their fine-grained differences. Note that even *EventTranR* with scrambled relations outperform ELMo with a large margin. We hypothesize that ELMo emphasizes similarity rather than nuanced discourse relations between sentences.

Methods	Accuracy	F1	MRR	Recall@4
Random	11.11	-	-	-
EventTransE	49.93	50.00	70.05	**83.05**
EventTransR	**50.84**	**51.00**	**70.62**	81.65

Table 5: Predicting relation type given two events, a 9-class classification task. **F1** is micro averaged.

	ELMo		EventTransE		EventTransR	
	Acc.	**F1**	**Acc.**	**F1**	**Acc.**	**F1**
Reason	60.82	59.02	70.04	69.65	69.35	67.46
Contrast	63.04	58.69	73.09	73.22	74.07	74.90
Cond.	61.92	60.48	71.16	71.22	71.46	72.25
Conj.	65.56	66.01	66.86	68.32	65.42	65.58
Result	60.68	60.84	73.26	73.41	72.69	72.98
Async.	61.25	60.59	71.25	69.61	73.22	**74.27**
Sync.	64.71	62.96	70.20	69.60	72.44	72.49
Instan.	62.64	58.84	**74.80**	**74.39**	73.30	71.83
Restat.	62.86	61.41	74.68	73.35	**75.37**	73.70
Average	62.61	60.98	71.70	71.42	**71.92**	**71.72**

Table 6: Triplet Classification for discourse relations.

Predict the Relation We predict the correct relation out of the 9 discourse relations (Table 1), given two events. Table 5 shows the result. With additional relation-specific parameters introduced, *EventTransR* performs better than *EventTransE*. Note that the ability to rank the correct relation is also important as there might be more than one possible next events. According to the MRR and Recall@4, both models are competitive.

Triplet Classification This task is inspired by Triplet Classifications in Knowledge Graph Completion (Socher et al., 2013; Wang et al., 2014). It predicts whether a given triplet (e_h, e_t, r) is valid or not. We sample positive triplets from our dev and test splits and negative triplets by corrupting e_t. We use the dev split to develop a set of relation-specific thresholds λ_r. The score is calculated using $f \in \{f_{trane}, f_{transr}\}$. If the score is lower than λ_r, the triplet is classified as positive; otherwise, it is negative. We sample 500 positive and negative triplets for each relation. The ELMo baseline is similar to previous experiments. We also develop a set of relation-specific thresholds based on ELMo's similarity scores to make predictions. Table 6 summarizes the results and shows that the similarity-based model, ELMo, cannot represent the nuanced relations information as good as our model. Interestingly, both our models excelled at predicting the *Expansion* relations (Instant. and Restat.). *EventTransR* get high scores on *Temporal* relations (Async.) which implies its applicability on tasks like event order inference (Ning et al., 2018). In general, for tasks requiring nuanced relations, *EventTransR* works better; if we only need to know the NEXT or COREF_NEXT events, *EventTransE* is better. In addition, *EventTransE* has less trainable parameters, converging way faster.

4.3 Implicit Discourse Sense Classifications

The final evaluation task is a subtask in CoNLL 2016 Shared Task (Xue et al., 2016) on implicit discourse sense classification. We follow the same setting as the shared task, with 15 sense classes. More details can be found in (Xue et al., 2016).

Three baselines, the best and median system of each subtask, are provided. In addition, we also trained a strong baseline based on ELMo. We first create word embeddings for words in the argument spans using ELMo and put an attention layer on top of the words. The attention layer weights the words and create the argument representation. We feed the representations of the two arguments to a neural classifier, where two fully-connected hidden layers with dimensions 256 and 128 are applied. ReLU (Nair and Hinton, 2010) are used as activation functions and AdaGrad (Duchi et al., 2011) is used for optimizing the parameters. We combine *EventTransE* with the ELMo baseline by having another attention layer on top of the event embeddings and concatenating all the argument representations in the network.

Methods	**Dev**	**Test**	**Blind**
PurdueNLP (Pacheco et al., 2016)	38.05	34.45	29.10
ecnucs (Wang and Lan, 2016)	46.42	40.91	34.18
ttr (Rutherford and Xue, 2016)	40.32	36.13	37.67
ELMo	45.60	37.65	36.72
ELMo+EventTransE	46.81	39.05	38.35

Table 7: micro F1 scores (%) for Implicit Discourse Sense Classifications. Evaluated against the best and median systems in CoNLL'16, and ELMo contextualized word embedding with attention layers, which can be improved by incorporating our *EventTransE*.

Table 7 shows the results. The ELMo baseline is highly competitive, comparable to the winners of the task (ecnucs and ttr). Our combined model (**ELMo+EventTransE**) consistently contributes to performance, demonstrating the benefit of our model to downstream tasks.

5 Summary

We consider the problem of learning relation-aware event embeddings for commonsense inference, which can account for different relations between events, beyond simple event similarity. We include several event relations, identifying, for example, the causes for them. We show that weak supervision, provided by a rule-based annotator is enough for training our models.

We evaluated and compared two models, *EventTransE* and *EventTransR*, on several narrative cloze and relation-specific tasks, and showed the learned embedding can capture relation-specific information as well as improve performance for a downstream task.

This work lays the foundation for reasoning over narratives and explaining how sentences combine to form them. In the future we would like to expand this direction, and find ways to connect event and relation representation, learning and inference in a unified framework.

Acknowledgements

We thank the anonymous reviewers of this paper for their insightful comments.

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under the ASED program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA.

References

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised learning of narrative event chains. In *ACL*, volume 94305, pages 789–797.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12(Jul):2121–2159.

Mark Granroth-Wilding and Stephen Clark. 2016. What happens next? event prediction using a compositional neural network model. In *AAAI*, pages 2727–2733.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Bram Jans, Steven Bethard, Ivan Vulić, and Marie Francine Moens. 2012. Skip n-grams and ranking functions for predicting script events. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 336–344. Association for Computational Linguistics.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *NIPS*, pages 3294–3302.

I-Ta Lee and Dan Goldwasser. 2018. Feel: Featured event embedding learning. *AAAI*, pages 4840–4847.

Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Constructing narrative event evolutionary graph for script event prediction. *arXiv preprint arXiv:1805.05081*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, volume 15, pages 2181–2187.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James F Allen. 2017. Lsdsem 2017 shared task: The story cloze test. *LSDSem 2017*, page 46.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814.

Maximilian Nickel, Lorenzo Rosasco, Tomaso A Poggio, et al. 2016. Holographic embeddings of knowledge graphs. In *AAAI*, volume 2, pages 3–2.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018. Joint reasoning for temporal and causal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2278–2288.

Maria Leonor Pacheco, I-Ta Lee, Xiao Zhang, Abdullah Khan Zehady, Pranjal Daga, Di Jin, Ayush Parolia, and Dan Goldwasser. 2016. Adapting event embedding for implicit discourse relation recognition. *Proceedings of the CoNLL-16 shared task*, pages 136–142.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition ldc2011t07. dvd. *Philadelphia: Linguistic Data Consortium*.

Haoruo Peng and Dan Roth. 2016. Two discourse driven language models for semantics. *arXiv preprint arXiv:1606.05679*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Karl Pichotta and Raymond J Mooney. 2014. Statistical script learning with multi-argument events. In *EACL*, volume 14, pages 220–229.

Karl Pichotta and Raymond J Mooney. 2016a. Learning statistical scripts with lstm recurrent neural networks. In *AAAI*.

Karl Pichotta and Raymond J Mooney. 2016b. Using sentence-level lstm language models for script inference. *arXiv preprint arXiv:1604.02993*.

Rashmi Prasad, Eleni Miltsakaki, Nikhil Dinesh, Alan Lee, Aravind Joshi, Livio Robaldo, and Bonnie L Webber. 2007. The penn discourse treebank 2.0 annotation manual.

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A Smith, and Yejin Choi. 2018. Event2mind: Commonsense inference on events, intents, and reactions. *arXiv preprint arXiv:1805.06939*.

Attapol Rutherford and Nianwen Xue. 2016. Robust non-explicit neural discourse parser in english and chinese. *Proceedings of the CoNLL-16 shared task*, pages 55–59.

Maarten Sap, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2018. Atomic: An atlas of machine commonsense for if-then reasoning. *arXiv preprint arXiv:1811.00146*.

Roger C Schank and Robert P Abelson. 1977. Scripts, plans, goals and understanding: An inquiry into human knowledge structures.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *NIPS*.

Andrew Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269.

Jianxiang Wang and Man Lan. 2016. Two end-to-end shallow discourse parsers for english and chinese in conll-2016 shared task. *Proceedings of the CoNLL-16 shared task*, pages 33–40.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, volume 14, pages 1112–1119.

Zhongqing Wang, Yue Zhang, and Ching-Yun Chang. 2017. Integrating order information and event relation for script event prediction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 57–67.

Noah Weber, Niranjan Balasubramanian, and Nathanael Chambers. 2017. Event representations with tensor-based compositions. *arXiv preprint arXiv:1711.07611*.

Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2016. Representation learning of knowledge graphs with hierarchical types. In *IJCAI*, pages 2965–2971.

Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Attapol Rutherford, Bonnie Webber, Chuan Wang, and Hongmin Wang. 2016. Conll 2016 shared task on multilingual shallow discourse parsing. *CoNLL-16 shared task*.

Hee-Geun Yoon, Hyun-Je Song, Seong-Bae Park, and Se-Young Park. 2016. A translation-based knowledge graph embedding preserving logical property of relations. In *NAACL*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.

Sendong Zhao, Quan Wang, Sean Massung, Bing Qin, Ting Liu, Bin Wang, and ChengXiang Zhai. 2017. Constructing and embedding abstract event causality networks from text snippets. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 335–344. ACM.

A Details of Event Preprocessing

This section notes the detailed preprocessing steps for extracting event predicate and arguments. It follows the previous work (Lee and Goldwasser, 2018), except the argument mention part.

- Unlike the previous works (Lee and Goldwasser, 2018; Granroth-Wilding and Clark, 2016; Pichotta and Mooney, 2016a), which only consider the headword of entity mentions, we use the entire mention span. This change gives the models a possibility to capture the nuanced information in the entity mentions, relevant for many commonsense inferences. For example, capturing the relationships between "a hungry man walked on a street" and "he grabbed some food" hinges on capturing the modifier "hungry."
- Predicates are lemmatized and in lower-case.
- Predicates are not only verbs but also predicative adjectives. For instance, "Jenny was hungry. She ordered a big meal." The predicate "hungry" plays an important role here.
- Negations should be applied to predicates, e.g., "She didn't eat dinner," results in a new predicate: "not_eat."
- Particles and clausal complements (xcomp) are included in verb predicates, since verbs, such as "go" and "have" are not strong enough to give meaningful information. For instance, in "He went shopping last night," the predicate is "go_shop," rather than "go."
- Low-frequency predicates and words in the entity mentions are considered as Out-Of-Vocabulry (OOV) during training. As the vocabulary size is related to memory limitation and rare words are highly likely to introduce noise, only the most active n_{pred} predicates and $n_{argword}$ argument words are considered.
- For the same reason given above, the maximum entity mention lengths, l_{subj} and l_{obj}, are set.

B Negative Sampling for Event Triplets

For each positive triplet (e_h, e_t, r), we extract one negative triplet by randomly replacing e_h, e_t, or r in equal chance. The events are sampled from event vocabulary, collected from the training set, and the relations are from the 11 types we support. We have experimented with different negative sampling strategies, such as corrupting the tail event only or sampling with different event distributions. None of them perform better.

C Hyperparameters

We have experimented with different sets of hyperparameters, and came up with the following setting: the number of active predicates and argument words, n_{pred} and $n_{argword}$, are both set to 25000; the maximum argument lengths, l_{subj} and l_{obj}, are set to 15; the event contextual window size $w_{context}$ for extracting NEXT relation is 5; the event composition hidden layer has the dimension $d_h = 1000$ and Rectified Linear Unit (ReLU) (Nair and Hinton, 2010) is used as the activation function.; embedding dimensions $d_a = 500, d_e = 500$, and $d_r = 500$; the margin δ is empirically set to 1; the optimizer is Adagrad (Duchi et al., 2011) with initial learning rate 0.01; the word embeddings for entity mention encoders are initialized as the word embeddings pre-trained in Skip-Thoughts (Kiros et al., 2015); all the experimental results are averaged over 5 runs.

D Qualitative Analysis

The experiments in the paper provide quantitative evaluations of our models. To give more comprehensive understanding, we also perform a qualitative analysis, which instantiates exact inferences our models make.

In this analysis, our models make inferences in grounded scenarios, where we have clearer expectations about possible events and outcomes. To do so, we create two confined "worlds," where each world only have limited numbers of entities and predicates, and hence a limited number of candidate events. This limitation is enforced as it helps examine quality of the inferences.

Table 8 shows the entities and predicates that are selected for the two worlds. The topic of the first world (a) is about a murderer and the topic of the second world (b) is about stock markets. Both are common topics in newswire articles, which we use for training the model. Note that since each event triplet has two entity components (subject and object) and one predicate, the number of candidate events is calculated as $n_{pred} \cdot n_{ent}^2$, where n_{pred} is number of predicates and n_{ent} is the number of entities. In these two worlds, we have 1100 and 1400 candidate events.

To conduct the inference, our model ranks the candidate events according to their relevance to a given starting scenario, which is a sequence of events. We use *EventTransR* to embed and rank all the events. For each candidate, we jointly consider its relevance to each start event. The dissimilarity scoring function $s(.)$ is defined as follows:

$$s(e_c) = \sum_{e_s \in S} f_{transr}(e_s, e_c, r), \quad \forall e_c \in C,$$

where S is all events in the starting scenario, C is the set of possible candidate events, and r is the embedding of the interested discourse relation. We rank all the candidates based on this function. Candidates with lower scores will be ranked higher. In addition, we consider four discourse relations—*Contrast*, *Reason*, *Result*, and *Asynchronous*—in this analysis, as they are particularly interesting for commonsense inference.

Table 9 summarizes the analysis. In each case, we only list the top 2-3 events. In world (a), *EventTransR* can precisely predict events in three out of four relations. In particular, we can contrast the fact that "John died" with "John survived," which has not been addressed in previous works. For *Asynchronous*, on which *EventTransR* fails, the signal for temporal relations is noisier as many possible outcomes are reasonable. In world (b), our model succeeds in all four relations. Also, our model is able to tell the difference between *Result* and *Reason*, as indicated by the prediction that "the stock has soared" *leads to* "CEO made money," and "*Because* shares increased, the stock soared." They show that we are able to control the inferences over different discourse perspectives, which is useful for tasks like story generations.

This analysis helps provide more intuitions about the knowledge learned by our models. Note that this is a challenging task even when grounded with a small set of candidate events, as was reported by previous works that looked at event-ranking based evaluations (Pichotta and Mooney, 2016a; Rashkin et al., 2018).

World	**Possible Entities**	**Possible Predicates**	**#Candidate Events**
(a)	jim, john, a girl the officer, he, a nurse, a man, a pedestrian, the gun, NOA	safe, hit, stop, hate, gone, happy, angry, smile, change, survive, sad	1100
(b)	shares, money, CEO, the price, CTO, police, employees, a dog, a girl, NOA	strike, hire, sell, buy, happy, sad, angry, smile, make, survive, adjust, increase, good, decrease	1400

Table 8: Small worlds for qualitative analysis: (a) murderer scenario; (b) stock market scenario. **NOA** stands for "No Argument."

	Triplets (a)	Interpretations (a)	Triplets (b)	Interpretations (b)
Starting Scenarios	(shot, jim, john), (die, john, NOA), (arrest, police, him)	Jim shot John. Police arrested him. Jim died.	(invest, company, fund), (have_soar, stock, NOA)	Company invested fund. The stock has soared.
Inference Contrast	**(survive, john, jim)**, **(survive, john, NOA)**	**John survived Jim.** **John survived.**	(increase, the price, ceo), **(strike, the price, shares)**	The price increased CEO. **The price stroke shares.**
Inference Result	**(sad, jim, NOA)**, (sad, john, jim)	**Jim was sad.** John sad Jim.	(increase, the price, ceo), **(make, ceo, shares)**, **(make, ceo, money)**	The price increases CEO. **CEO made shares.** **CEO made money.**
Inference Reason	(angry, NOA, john), **(angry, jim, NOA)**	__ angry John. **Jim was angry.**	**(increase, shares, NOA)**, (buy, employees, ceo)	**Shares increased.** Employees bought CEO.
Inference Async.	(survive, NOA, john), (survive, jim, john)	__ survived John. Jim survived John.	**(make, shares, ceo)**, **(make, money, ceo)**	**CEO made shares.** **CEO made money.**

Table 9: Qualitative Analysis: two worlds are given, where world (a) is shown in column 2 and 3, and world (b) is shown in column 4 and 5. World (a) has 1100 and world (b) has 1400 number of possible event candidates respectively. The first row, starting scenarios, give the start events and the below 4 rows show the inference based on 4 discourse relations that we are particularly interested in. Events that match commonsense are bolded. **NOA** stands for "No Argument."