Learning Fine-Grained Knowledge about Contingent Relations between Everyday Events

Elahe Rahimtoroghi, Ernesto Hernandez and Marilyn A Walker
Natural Language and Dialogue Systems Lab
Department of Computer Science, University of California Santa Cruz
Santa Cruz, CA 95064, USA
elahe@soe.ucsc.edu, eherna23@ucsc.edu, mawalker@ucsc.edu

Abstract

Much of the user-generated content on social media is provided by ordinary people telling stories about their daily lives. We develop and test a novel method for learning fine-grained common-sense knowledge from these stories about contingent (causal and conditional) relationships between everyday events. This type of knowledge is useful for text and story understanding, information extraction, question answering, and text summarization. We test and compare different methods for learning contingency relation, and compare what is learned from topic-sorted story collections vs. general-domain stories. Our experiments show that using topic-specific datasets enables learning finer-grained knowledge about events and results in significant improvement over the baselines. An evaluation on Amazon Mechanical Turk shows 82% of the relations between events that we learn from topic-sorted stories are judged as contingent.

1 Introduction

The original idea behind scripts as introduced by Schank was to capture knowledge about the fine-grained events of everyday experience, such as opening a fridge enabling preparing food, or the event of getting out of bed being triggered by an alarm going off (Schank and Abelson, 1977; Mooney and DeJong, 1985) This idea has motivated previous work exploring whether common-sense knowledge about events can be learned from text, however, only a few learn from data other than newswire (Hu et al., 2013; Manshadi et al., 2008; Beamer and Girju, 2009). News articles (obviously) cover newsworthy topics such as bombing, explosions, war and killing so the knowledge learned is limited to those types of events.

However, much of the user-generated content on social media is provided by ordinary people telling stories about their daily lives. These stories are rich with common-sense knowledge. For example, the Camping Trip story in Fig. 1 contains implicit common-sense knowledge about contingent (causal and conditional) relations between camping-related events, such as setting up a tent and placing a tarp. The Storm story contains implicit knowledge about events such as the hurricane made landfall, the wind blew, a tree fell. Our aim is to learn fine-grained common-sense knowledge about contingent relations between everyday events from such stories. We show that the fine-
grained knowledge we learn is simply not found in publicly available narrative and event schema collections (Chambers and Jurafsky, 2009; Balasubramanian et al., 2013).

Personal stories provide both advantages and disadvantages for learning common-sense knowledge about events. An advantage is that they tend to be told in chronological order (Swanson and Gordon, 2009), and temporal order between events is a strong cue to contingency (Prasad et al., 2008; Beamer and Girju, 2009). However, their structure is more similar to oral narrative than to newswire (Rahimtoroghi et al., 2014; Swanson et al., 2014).

Only about a third of the sentences in a personal narrative describe actions,\(^1\) so novel methods are needed to find useful relationships between events. Another difference between our work and prior research is that much of the work on narrative schemas, scripts, or event schemas characterize what is learned as “collections of events that tend to co-occur”. Thus what is learned is not evaluated for contingency (Chambers and Jurafsky, 2008; Chambers and Jurafsky, 2009; Manshadi et al., 2008; Nguyen et al., 2015; Balasubramanian et al., 2013; Pichotta and Mooney, 2014). Historically, work on scripts explicitly modeled causality (Lehnert, 1981; Mooney and DeJong, 1985) *inter alia*. Our work is motivated by Penn Discourse Treebank (PDTB) definition of CONTINGENCY that has two types: CAUSE and CONDITION, and is more similar to approaches that learn specific event relations such as contingency or causality (Hu et al., 2013; Do et al., 2011; Girju, 2003; Riaz and Girju, 2010; Rink et al., 2010; Chklovski and Pantel, 2004). Our contributions are as follows:

- We use a corpus of everyday events for learning common-sense knowledge focusing on the contingency relation between events. We first use a subset of the corpus including general-domain stories. Next, we produce a topic-sorted set of stories using a semi-supervised bootstrapping method to learn finer-grained knowledge. We use two different datasets to directly compare what is learned from topic-sorted stories as opposed to a general-domain story corpus (Sec. 2);

- We develop a new method for learning contingency relations between events that is tailored to the “oral narrative” nature of blog stories. We apply Causal Potential (Beamer and Girju, 2009) to model the contingency relation between two events. We directly compare our method to several other approaches as baselines (Sec. 3). We also identify topic-indicative contingent event pairs from our topic-specific corpus that can be used as building blocks for generating coherent event chains and narrative schema for a particular theme (Sec. 4.3);

- We conduct several experiments to evaluate the quality of the event knowledge learned in our work that indicate our results are contingent and topic-related. We directly compare the common-sense knowledge we learn with the Rel-grams collection and show that what we learn is not found in available corpora (Sec. 4).

We release our contingent event pair collections for each topic for future use of other research groups.\(^2\)

2 A Corpus of Everyday Events

Our dataset is drawn from the Spinn3r corpus of millions of blog posts (Burton et al., 2009; Gordon and Swanson, 2009; Gordon et al., 2012). We hypothesize that personal stories are a valuable resource to learn common-sense knowledge about relations between everyday events and that finer-grained knowledge can be learned from topic-sorted stories (Riaz and Girju, 2010) that share a particular theme, so we construct two different sets of stories:

**General-Domain Set.** We created a random subset from the Spinn3r corpus from personal blog domains: *livejournal.com*, *wordpress.com*, *blogspot.com*, *spaces.live.com*, *typepad.com*, *travelpod.com*. This set consists of 4,200 stories not selected for any specific topic.

**Topic-Specific Set.** We produced a dataset by filtering the corpus using a bootstrapping method to create topic-specific sets for topics such as going camping, being arrested, going snorkeling or scuba diving, visiting the dentist, witnessing a major storm, and holiday activities associated with Thanksgiving and Christmas (see Table 1).

We apply AutoSlog-TS, a semi-supervised algorithm that learns narrative *event-patterns* to bootstrap a collection of stories on the same

\(^1\)The other two thirds provide scene descriptions and descriptions of the thoughts or feelings of the narrator.

\(^2\)https://nlds.soe.ucsc.edu/everyday_events
Table 1: Some topics and examples of their indicative events.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camping Trip</td>
<td>camp(), roast(dobj:marshmallow), hike(), pack(), fish(), go(dobj:camp), grill(), put(dobj:tent, prt:up), build(dobj:fire)</td>
</tr>
<tr>
<td>Storm</td>
<td>restore(), lose(dobj:power), rescue(), evacuate(), flood(), damage(), sustain(), survive(), watch(dobj:storm)</td>
</tr>
<tr>
<td>Christmas Holidays</td>
<td>open(dobj:present), wrap(), celebrate(), sing(), play(), exchange(dobj:gift), snow(), buy(), decorate(dobj:tree)</td>
</tr>
<tr>
<td>Snorkeling and Scuba Diving</td>
<td>see(dobj:fish), swim(), snorkel(), sail(), surface(), dive(), dart(), rent(dobj:equipment), enter(dobj:water), see(dobj:turtle)</td>
</tr>
</tbody>
</table>

Table 2: Examples of narrative event-patterns (case frames) learned from corpus.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Event-Pattern (Case Frame) Examples</th>
</tr>
</thead>
</table>

1. Hand-labeling: We manually labeled a small set (∼200-300) of stories on the topic.
2. Generating Event-Patterns: Given hand-labeled stories on a topic (from Step 1), and a random set of stories that are not relevant to that topic, AutoSlog-TS learns a set of syntactic templates (case frame templates) that distinguish the linguistic patterns characteristic of the topic from the random set. For each pattern it generates frequency and conditional probability which indicate how strongly the pattern is associated with the topic.

3. Parameter Tuning: We use the frequency and probability generated by AutoSlog-TS and apply a threshold for filtering to select a subset of indicative event-patterns strongly associated with the topic. In this step we aim to find optimal values for frequency and probability thresholds denoted as \( f\)-threshold and \( p\)-threshold respectively. We divided the hand-labeled data from Step 1 into train and development sets and designed a classifier based on our bootstrapping method: if the number of event-patterns extracted from a post is more than a certain number (\( n\)-threshold), it is labeled as positive and otherwise it is labeled as negative meaning that it is not related to the topic. We repeated the classification for several combinations of different values for each of the three parameters and measured the precision, recall and f-measure. We selected the optimal values for the thresholds that resulted in high precision (above 0.9) and average recall (around 0.4). We compromised on a lower recall to achieve a high precision to establish a highly accurate bootstrapping algorithm. Since bootstrapping is performed on a large set of stories, a low recall stills result in identifying enough stories per topic.

4. Bootstrapping: We use the patterns learned in previous steps as indicative event-patterns for the topic. The bootstrapping algorithm processes each story, using AutoSlog-TS to extract lexico-syntactic patterns. Then it counts the indicative event-patterns in the extracted patterns, and labels the blog as a positive instance for that topic if the count is above the \( n\)-threshold value for that topic.

The manually labeled dataset includes 361 Storm and 299 Camping Trip stories. After one round of bootstrapping the algorithm identified 971 additional Storm and 870 more Camping Trip stories. The bootstrapping method is not evaluated separately, however, the results in Sec. 4.2 indicate that using the bootstrapped data considerably improves the accuracy of the contingency model and enhances extracting topic-relevant event knowledge.
3 Learning Contingency Relation between Narrative Events

In this section we describe our representation of events in narratives and our methods for modeling contingency relationship between events.

3.1 Event Representation

In previous work different representations have been proposed for the event structure such as single verb and verb with two or more arguments. Verbs are used as a central indication of an event in a narrative. However, other entities related to the verb also play a strong role in conveying the meaning of the event. In (Pichotta and Mooney, 2014) it is shown that the multi-argument representation is richer than the previous ones and is capable of capturing interactions between multiple events. We use a representation that incorporates the Particle of the verb in the event structure in addition to the Subject and the Direct Object and define an event as a verb with its dependency relations as follows:

Verb Lemma (subj:Subject Lemma, dobj:Direct Object Lemma, prt:Particle)

Table 3 shows example sentences describing an event from the Camping topic along with their event structure. The examples show how including the arguments often change the meaning of an event. In Row 1 the direct object and particle are required to completely understand the event in this sentence. Row 2 shows another example where the verb have cannot implicate what event is happening and the direct object oatmeal is needed to understand what has occurred in the story.

We parse each sentence and extract every verb lemma with its arguments using Stanford dependencies (Manning et al., 2014). For each verb, we extract the nsubj, dobj, and prt dependency relations if they exist, and use their lemma in the event representation. To generalize the event representations, we use the types identified by Stanford’s Named Entity Recognizer and map each argument to its named entity type if available, e.g., in Row 3 of Table 3, the Lost River Valley Campground is represented by its type LOCATION. We use abstract types for named entities such as PERSON, ORGANIZATION, TIME and DATE. We also represent each pronoun by the abstract type PERSON, e.g. Row 5 in Table 3.

3.2 Causal Potential Method

We define a contingent event pair as a sequence of two events \((e_1, e_2)\) such that \(e_1\) and \(e_2\) are likely to occur together in the given order and \(e_2\) is contingent upon \(e_1\). We apply an unsupervised distributional measure called Causal Potential to induce the contingency relation between two events.

Causal Potential (CP) was introduced by Beamer and Girju (2009) as a way to measure the tendency of an event pair to encode a causal relation, where event pairs with high CP have a higher probability of occurring in a causal context. We calculate CP for every pair of adjacent events in each topic-specific dataset. We used a 2-skip bigram model which considers two events to be adjacent if the second event occurs within two or less events after the first one.

We use skip-2 bigram in order to capture the fact that two related events may often be separated by a non-essential event, because of the oral-narrative nature of our data (Rahimtoroghi et al., 2014). In contrast to the verbs that describe an event (e.g., hike, climb, evacuate, drive), some verbs describe private states such as as belong, depend, feel, know. We filter out clauses that tend to be associated with private states (Wiebe, 1990). A pilot evaluation showed that this improves the results.

Equation 1 shows the formula for calculating Causal Potential of a pair consisting of two events: \((e_1, e_2)\). Here \(P\) denotes probability and \(P(e_1 \rightarrow e_2)\) is the probability of \(e_2\) occurring after \(e_1\) in the adjacency window which is equal to 3 due to the skip-2 bigram model. \(P(e_2|e_1)\) is the conditional probability of \(e_2\) given that \(e_1\) has been seen in the adjacency window. This is equivalent to the
Event-Bigram model described in Sec. 3.3.

\[
CP(e_1, e_2) = \log \frac{P(e_2|e_1)}{P(e_2)} + \log \frac{P(e_1 \rightarrow e_2)}{P(e_2 \rightarrow e_1)}
\]

(1)

To calculate CP, we need to compute event counts from the corpus and thus we need to define when two events are considered equal. The simplest approach is to define two events to be equal when their verb and arguments exactly match. However, with a close look at the data this approach does not seem adequate. For example, consider the following events:

- go (subj:PERSON, dobj:camp)
- go (subj:family, dobj:camp)
- go (dobj:camp)

They encode the same action although their representations do not exactly match and differ in the subject. Our intuition is that when we count the number of events represented as go (subj:PERSON, dobj:camp) we should also include the count of go (dobj:camp). To be able to generalize over the event structure and take into account these nuances, we consider two events to be equal if they have the same verb lemma and share at least one argument other than the subject.

### 3.3 Baseline Methods

Our previous work on modeling contingency relations in film scripts data compared Causal Potential to methods used in previous work: Bigram event models (Manshadi et al., 2008) and Pointwise Mutual Information (PMI) (Chambers and Jurafsky, 2008) and the evaluations showed that CP obtains better results (Hu et al., 2013). In this work, we use CP for inducing contingency relation between events and apply three other models as baselines for comparison:

- **Event-Unigram.** This method will produce a distribution of normalized frequencies for events.
- **Event-Bigram.** We calculate the bigram probability of every pair of adjacent events using skip-2 bigram model using the Maximum Likelihood Estimation (MLE) from our datasets:

\[
P(e_2|e_1) = \frac{\text{Count}(e_1, e_2)}{\text{Count}(e_1)}
\]

(2)

- **Event-SCP.** We use the Symmetric Conditional Probability between event tuples (Rel-grams) used in (Balasubramanian et al., 2013) as another baseline method. The Rel-gram model is the most relevant previous work to our method and outperforms the previous state of the art on generating narrative event schema. This metric combines bigram probability considering both directions:

\[
SCP(e_1, e_2) = P(e_2|e_1) \times P(e_1|e_2)
\]

(3)

Like Event-Bigram, we used MLE for estimating Event-SCP from the corpus.

### 4 Evaluation Experiments

We conducted three sets of experiments to evaluate different aspects of our work. First, we compare the content of our topic-specific event pairs to current state of the art event collections to show that the fine-grained knowledge we learned about everyday events does not exist in previous work focused on the news genre. Second, we run an automatic evaluation test, modeled after the COPA task (Roemmele et al., 2011), on a held-out test set to evaluate the event pair collections that we have extracted from both General-Domain and Topic-Specific datasets, in terms of contingency relations. We hypothesize that the contingent event pairs can be used as basic elements for generating coherent event chains and narrative schema. So, in the third part of the experiments, we extract topic-indicative contingent event pairs from our Topic-Specific dataset and run an experiment on Amazon Mechanical Turk (AMT) to evaluate the top N pairs with respect to their contingency relation and topic-relevance.

### 4.1 Comparison to Rel-gram Tuple Collections

We chose Rel-gram tuples (Balasubramanian et al., 2013) for comparison since it is the most relevant previous work to us: they generate pairs of relational tuples of events, called Rel-grams using co-occurrence statistics based on Symmetric Conditional Probability described in Sec 3.3. Additionally, the Rel-grams are publicly available

<table>
<thead>
<tr>
<th>Label</th>
<th>Rel-gram Tuples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contingent &amp; Strongly Relevant</td>
<td>7 %</td>
</tr>
<tr>
<td>Contingent &amp; Somewhat Relevant</td>
<td>0 %</td>
</tr>
<tr>
<td>Contingent &amp; Not Relevant</td>
<td>35 %</td>
</tr>
<tr>
<td>Total Contingent</td>
<td>42 %</td>
</tr>
</tbody>
</table>

Table 4: Evaluation of Rel-gram tuples on AMT.
through an online search interface and their evaluations show that their method outperforms the previous state of the art on generating narrative event schema.

However, their work is focused on news articles and does not consider the causal relation between events for inducing event schema. We compare the content of what we learned from our topic-specific corpus to the Rel-gram tuples to show that the fine-grained type of knowledge that we learn is not found in their events collection. We also applied the co-occurrence statistics that they used on our data as a baseline (Event-SCP) for comparison to our method and present the results in Sec. 4.2.

In this experiment we compare the event pairs extracted from our Camping Trip topic to the Rel-gram tuples. The Rel-gram tuples are not sorted by topic. To find tuples relevant to Camping Trip, we used our top 10 indicative events and extracted all the Rel-gram tuples that included at least one event corresponding to one of the Camping Trip indicative events. For example, for go(dobj:camp), we pulled out all the tuples that included this event from the Rel-grams collection. The indicative events for each topic were automatically generated during the bootstrapping using AutoSlog-TS (Sec. 2).

Then we applied the same sorting and filtering methods presented in the Rel-grams work and removed any tuple with frequency less than 25 and sorted the rest by the total symmetrical conditional probability. These numbers are publicly available as a part of the Rel-grams collection. We evaluated the top $N = 100$ tuples of this list using the Mechanical Turk task described later in Sec. 4.3. The evaluation results presented in Table 4 show that 42% of the Rel-gram pairs were labeled as contingent by the annotators and only 7% were both contingent and topic-relevant. We argue that this is mainly due to the limitations of the newswire data which does not contain the fine-grained everyday events that we have extracted from our corpus.

### 4.2 Automatic Two-Choice Test

For evaluating our contingent event pair collections we have automatically generated a set of two-choice questions along with the answers, modeled after the COPA task (Roemmele et al., 2011). We produced questions from held-out test sets for each dataset. Each question consists of one event and two choices. The question event is one that occurs in the test data. One of the choices is an event adjacent to the question event in the document. The other choice is an event randomly selected from the list of all events occurring in the test set. The following is an example of a question from the Camping Trip test set:

**Question event:** arrange (dobj:outdoor)

**Choice 1:** help (dobj:trip)

**Choice 2:** call (subj:PERSON)

In this example, arrange (dobj:outdoor) is followed by the event help (dobj:trip) in a document from the test set and call (subj:PERSON) was randomly generated. The model is supposed to predict which of the two choices is more likely to have a contingency relation with the event in the question. We argue that a strong contingency model should be able to choose the correct answer (the one that is adjacent to the question event) and the accuracy achieved on the test questions is an indication of the model’s robustness.

For the General-Domain dataset, we split the data into train (4,000 stories) and held-out test (200 stories) sets. For each topic-specific set, we divided the hand-labeled data into a train (Train-HL) and held-out test, and created a second train set consisting of Train-HL and the data collected by bootstrapping (Train-HL-BS) as shown in Table 5. We automatically created a question for every event occurring in the test data which

---

### Table 5: Number of stories in the train and test sets from topic-specific dataset.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Dataset</th>
<th># Docs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camping Trip</td>
<td>Hand-labeled held-out test</td>
<td>107</td>
</tr>
<tr>
<td>Storm</td>
<td>Hand-labeled held-out test</td>
<td>98</td>
</tr>
<tr>
<td>Camping Trip</td>
<td>Hand-labeled train (Train-HL)</td>
<td>192</td>
</tr>
<tr>
<td>Storm</td>
<td>Hand-labeled train (Train-HL)</td>
<td>263</td>
</tr>
<tr>
<td>Camping Trip</td>
<td>Train-HL + Bootstrap (Train-HL-BS)</td>
<td>1,062</td>
</tr>
<tr>
<td>Storm</td>
<td>Train-HL + Bootstrap (Train-HL-BS)</td>
<td>1,234</td>
</tr>
</tbody>
</table>

### Table 6: Automatic two-choice test results for General-Domain dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event-Unigram</td>
<td>0.478</td>
</tr>
<tr>
<td>Event-Bigram</td>
<td>0.481</td>
</tr>
<tr>
<td>Event-SCP (Rel-gram)</td>
<td>0.477</td>
</tr>
<tr>
<td>Causal Potential</td>
<td>0.510</td>
</tr>
</tbody>
</table>

---

3http://relgrams.cs.washington.edu:10000/relgrams
Table 7: Automatic two-choice test results for Topic-Specific dataset.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Model</th>
<th>Train Dataset</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camping</td>
<td>Event-Unigram</td>
<td>Train-HL-BS</td>
<td>0.507</td>
</tr>
<tr>
<td>Trip</td>
<td>Event-Bigram</td>
<td>Train-HL-BS</td>
<td>0.510</td>
</tr>
<tr>
<td></td>
<td>Event-SCP</td>
<td>Train-HL-BS</td>
<td>0.508</td>
</tr>
<tr>
<td></td>
<td>Causal Potential</td>
<td>Train-HL</td>
<td>0.631</td>
</tr>
<tr>
<td></td>
<td>Causal Potential</td>
<td>Train-HL-BS</td>
<td>0.685</td>
</tr>
<tr>
<td>Storm</td>
<td>Event-Unigram</td>
<td>Train-HL-BS</td>
<td>0.510</td>
</tr>
<tr>
<td></td>
<td>Event-Bigram</td>
<td>Train-HL-BS</td>
<td>0.523</td>
</tr>
<tr>
<td></td>
<td>Event-SCP</td>
<td>Train-HL-BS</td>
<td>0.516</td>
</tr>
<tr>
<td></td>
<td>Causal Potential</td>
<td>Train-HL</td>
<td>0.711</td>
</tr>
<tr>
<td></td>
<td>Causal Potential</td>
<td>Train-HL-BS</td>
<td>0.887</td>
</tr>
</tbody>
</table>

Table 8: Results of evaluating indicative contingent event pairs on AMT.

<table>
<thead>
<tr>
<th>Label</th>
<th>Camping</th>
<th>Storm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contingent &amp; Strongly Relevant</td>
<td>44 %</td>
<td>33 %</td>
</tr>
<tr>
<td>Contingent &amp; Somewhat Relevant</td>
<td>8 %</td>
<td>20 %</td>
</tr>
<tr>
<td>Contingent &amp; Not Relevant</td>
<td>30 %</td>
<td>24 %</td>
</tr>
<tr>
<td>Total Contingent</td>
<td>82 %</td>
<td>77 %</td>
</tr>
</tbody>
</table>

Fig. 2 shows some examples of event pairs with high CP scores extracted from general-Domain set. In the following section we extract topic-indicative contingent event pairs and show that Topic-Specific data enables learning of finer-grained event knowledge that pertain to a particular theme.

4.3 Topic-Indicative Contingent Event Pairs

We identify contingent event pairs that are highly indicative of a particular topic. We hypothesize that these event pairs serve as building blocks of coherent event chains and narrative schema since they encode contingency relation and correspond to a specific theme. We evaluate the pairs on Amazon Mechanical Turk (AMT).

To identify event sequences that have a strong correlation to a topic (topic-indicative pairs) we applied two filtering methods. First, we selected the frequent pairs for each topic and removed the ones that occur less than 5 times in the corpus. Second, we used the indicative event-patterns for each topic and extracted the pairs that at least included one of these patterns. Indicative event-patterns are automatically generated during the bootstrapping using AutoSlog-TS and mapped to their corresponding event representation as described in Sec. 2. Then we used the Causal Potential scores from our contingency model for ranking the topic-indicative event pairs to identify the highly contingent ones. We sorted the pairs based on the Causal Potential score and evaluated the top N pairs in this list.

Evaluations and Results. We evaluate the indicative contingent event pairs using human judgment on Amazon Mechanical Turk (AMT). Narrative schema consists of chains of events that are related in a coherent way and correspond to a common theme. Consequently, we evaluate the extracted pairs based on two main criteria:

- Contingency: Two events in the pair are
### Table 9: Examples of event pairs evaluated on AMT.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Label &gt; 2 : Contingent &amp; Strongly Topic-Relevant</th>
<th>Label &lt; 1 : Not Contingent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camping Trip</td>
<td>person - pack up → person - go - home</td>
<td>person - pick up - cup → person - swim</td>
</tr>
<tr>
<td></td>
<td>person - wake up → person - pack up - backpack</td>
<td>pack up - tent → check out - video</td>
</tr>
<tr>
<td></td>
<td>person - head → hike up</td>
<td>person - play → person - pick up - sax</td>
</tr>
<tr>
<td></td>
<td>climb → person - find - rock</td>
<td>pack up - material → switch off - projector</td>
</tr>
<tr>
<td></td>
<td>person - pack up - car → head out</td>
<td>person - pick up - photo → person - swim</td>
</tr>
<tr>
<td>Storm</td>
<td>wind - blow - transformer → power - go out</td>
<td>restore - community → hurricane - bend</td>
</tr>
<tr>
<td></td>
<td>tree - fall - cave → crush</td>
<td>boil → tree - fall - driveway</td>
</tr>
<tr>
<td></td>
<td>like - blow → knock down - limb</td>
<td>clean up - person → people - come out</td>
</tr>
<tr>
<td></td>
<td>air - push - person → person - fall out</td>
<td>blow - sign → person - sit</td>
</tr>
<tr>
<td></td>
<td>hit - location → evacuate - person</td>
<td>person - rock - way → bottle - fall</td>
</tr>
</tbody>
</table>

likely to occur together in the given order and the second event is contingent upon the first one.

- **Topic Relevance:** Both events strongly correspond to the specified topic.

We have designed one task to assess both criteria since if an event pair is not contingent, it cannot be used in narrative schema for not satisfying the required coherence (even if it is topic-relevant). We asked the AMT annotators to rate each pair on a scale of 0-3 as follows:

- **0:** The events are not contingent.
- **1:** The events are contingent but not relevant to the specified topic.
- **2:** The events are contingent and somewhat relevant to the specified topic.
- **3:** The events are contingent and strongly relevant to the specified topic.

To ensure that the Amazon Mechanical Turk annotations are reliable, we designed a Qualification Type which requires the workers to pass a test before they can annotate our pairs. If the workers score 70% or more on the test they will qualify to do the main task. For each topic we created a Qualification test consisting of 10 event pairs from that topic that were annotated by two experts. To make the events more readable for the annotators we used the following representation:

- **Subject - Verb Particle - Direct Object**

For example, `hike(subj:person, dobj:trail, prt:up)` is mapped to `person - hike up - trail`. For each topic we evaluated top $N = 100$ event pairs and assigned 5 workers to rate each one. We generated a gold standard label for each pair by averaging over the scores assigned by the annotators and interpreted the average as follows:

- **Label >2:** Contingent & strongly topic-relevant.
- **Label = 2:** Contingent & somewhat topic-relevant.
- **1 ≤ Label < 2:** Contingent & not topic-relevant.
- **Label < 1:** Not contingent.

To assess the inter-annotator reliability we calculated kappa between each worker and the majority of the labels assigned to each pair. The average kappa was 0.73 which indicates substantial agreement. The results in Table 8 show that 52% of the Camping Trip and 53% of the Storm pairs were labeled as contingent and topic-relevant by the annotators. The results also indicate that our model is capable of identifying event pairs with strong contingency relations: 82% of the Camping Trip pairs and 77% of the Storm pairs were marked as contingent by the workers. Examples of the strongest and weakest pairs evaluated on Mechanical Turk are shown in Table 9. By comparison to Fig. 2, we can see that we can learn finer-grained type of events knowledge from topic-specific stories as compared to general-domain corpus.

### 5 Discussion and Conclusions

We learned fine-grained common-sense knowledge about contingent relations between everyday events from personal stories written by ordinary people. We applied a semi-supervised bootstrapping approach using event-patterns to create topic-sorted sets of stories and evaluated our methods on a set of general-domain narratives as well as two topic-specific datasets. We developed a new method for learning contingency relations between events that is tailored to the “oral narrative” nature of the blog stories. Our evaluations indi-
cate that a method that works well on the news genre does not generate coherent results on personal stories (comparison of Event-SCP baseline with Causal Potential).

We modeled the contingency (causal and conditional) relation between the events from each dataset using Causal Potential and evaluated on the questions automatically generated from a held-out test set. The results show significant improvement over the Event-Unigram, Event-Bigram, and Event-SCP (Rel-grams method) baselines on Topic-Specific stories: 25% improvement of accuracy on Camping Trip and 41% on Storm topic compared to Bigram model. In our future work, we plan to explore existing topic-modeling algorithms to create a broader set of topic-sorted corpora for learning contingent event knowledge.

Our experiments show that most of the fine-grained contingency relations we learn from narrative events are not found in existing narrative and event schema collections induced from the newswire datasets (Rel-grams). We also extracted indicative contingent event pairs from each topic and evaluated them on Mechanical Turk. The evaluations show that 82% of the relations between events that we learn from topic-sorted stories are judged as contingent. We publicly release the extracted pairs for each topic. In future work, we plan to use the contingent event pairs as building blocks for generating coherent event chains and narrative schema on several different themes.

References


Andrew Gordon and Reid Swanson. 2009. Identifying personal stories in millions of weblog entries. In Third International Conference on Weblogs and Social Media, Data Challenge Workshop, San Jose, CA.


Elahe Rahimtoroghi, Thomas Corcoran, Reid Swanson, Marilyn A. Walker, Kenji Sagae, and Andrew S. Gordon. 2014. Minimal narrative annotation schemes and their applications. In 7th Workshop on Intelligent Narrative Technologies.


Bryan Rink, Cosmin Adrian Bejan, and Sanda M Harabagiu. 2010. Learning textual graph patterns to detect causal event relations. In FLAIRS Conference.


