Weakly Supervised Subevent Knowledge Acquisition

Wenlin Yao[1] **Zeyu Dai**[1] **Maitreyi Ramaswamy**[1] **Bonan Min**[2] **Ruihong Huang**[1]
[1] Department of Computer Science and Engineering, Texas A&M University
[2] Raytheon BBN Technologies
(wenlinyao,jzdaizeyu,maitreyiramaswam,huangrh)@tamu.edu
bonan.min@raytheon.com

Abstract

Subevents elaborate an event and widely exist in event descriptions. Subevent knowledge is useful for discourse analysis and event-centric applications. Acknowledging the scarcity of subevent knowledge, we propose a weakly supervised approach to extract subevent relation tuples from text and build the first large scale subevent knowledge base. We first obtain the initial set of event pairs that are likely to have the subevent relation, by exploiting two observations that 1) subevents are temporally contained by the parent event, and 2) the definitions of the parent event can be used to further guide the identification of subevents. Then, we collect rich weak supervision using the initial seed subevent pairs to train a contextual classifier using BERT and apply the classifier to identify new subevent pairs. The evaluation showed that the acquired subevent tuples (239K) are of high quality (90.1% accuracy) and cover a wide range of event types. The acquired subevent knowledge has been shown useful for discourse analysis and identifying a range of event-event relations[1].

1 Introduction

A subevent is the event that happens as a part of the other event (i.e., parent event) spatio-temporally (Glavaš and Šnajder, 2014). Subevents, which elaborate and expand an event, widely exist in event descriptions. For instance, when describing *election* events, people usually describe typical subevents such as "*nominate* candidates", "*debates*" and "people *vote*". Knowing typical subevents of an event can help with analyzing several discourse relations (such as expansion and temporal relations) between text units. Furthermore, knowing typical subevents of an event is important for understanding the internal structure of the event (what is the event about?) and its properties (is this a violent or peaceful event?), and therefore has great potential to benefit event detection, event tracking, event visualization and event summarization among many other applications.

While being in high demand, little subevent knowledge can be found in existing knowledge bases. Therefore, we aim to extract subevent knowledge from text and build the first subevent knowledge base covering a large number of commonly seen events and their rich subevents.

Little research has focused on identifying the subevent relation between two events in a text. Several datasets annotated with subevent relations (Glavaš et al., 2014; Araki et al., 2014; O'Gorman et al., 2016) exist, but they are extremely small and usually contain dozens to one/two hundred documents. Subevent relation classifiers trained on these small datasets are not suitable to use to extract subevent knowledge from text, considering that subevent relations can appear in dramatically different contexts depending on topics and events.

We propose to conduct weakly supervised learning and train a wide-coverage contextual classifier to acquire diverse event pairs of the subevent relation from text. We start by creating **weak supervision**, where we aim to identify the initial set of subevent relation tuples from a text corpus. With no contextual classifier at the beginning, it is difficult to extract subevent relation tuples because subevent relations are rarely stated explicitly. Instead, we propose a novel two-step approach to indirectly obtain the initial set of subevent relation tuples, exploiting two key observations that (1) subevents are temporally contained by the parent event, and thus can be extracted with linguistic expressions that in-

[1]Code and the knowledge base are available at https://github.com/wenlinyao/EMNLP20-SubeventAcquisition

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 5345–5356,
November 16–20, 2020.

dicate the temporal containment relationship[2], and (2) the definition of the parent event is useful to prune spurious subevent tuples away to improve the quality.

Specifically, we first use several preposition patterns (e.g., e_i *during* e_j) that indicate the temporal relation *contained_by* between events to identify candidate subevent relation tuples. Then, we conduct an event definition-guided semantic consistency check to remove spurious subevent tuples that often include two temporally overlapping but semantically incompatible events. For example, a news article may report a *bombing* event that happened in parallel during a *festival*, but the intense *bombing* event is not semantically compatible with the entertaining event *festival*, as informed by the common definition of *festival*:

> *A festival is an organized series of celebration events, or an organized series of concerts, plays, or movies, typically one held annually.*

Next, we identify sentences from the text corpus that contain an event pair, and use these sentences to train a contextual classifier that can recognize the subevent relation in text. We train the contextual subevent relation classifier by fine-tuning the pre-trained BERT model (Devlin et al., 2019). We then apply the contextual BERT classifier to identify new event pairs that have the subevent relation.

We have built a large knowledge base of 239K subevent relation tuples. The knowledge base contains subevents for 10,318 unique events, with each event associated with 20.1 subevents on average. Intrinsic evaluation demonstrates that the learned subevent relation tuples are of high quality (90.1% of accuracy) and are valuable for event ontology building and exploitation.

The learned subevent knowledge has been shown useful for identifying subevent relations in text, including both intra-instance and cross-sentence cases. In addition, the learned subevent knowledge is shown useful for identifying temporal and causal relations between events as well, for the challenging cross-sentence cases where we usually have little contextual clues to rely on. Furthermore, when incorporated into a recent neural discourse parser, the learned subevent knowledge has noticeably improved the performance for identifying two types of implicit discourse relations, expansion and temporal relations.

In short, we made three main contributions: 1). We developed a novel weakly supervised approach to acquire subevent knowledge from text. 2). We have built the first large scale subevent knowledge base that is of high quality and covers a wide range of event types. 3). We performed extensive evaluation showing that the harvested subevent knowledge not only improves subevent relation extraction, but also improves a wide range of NLP tasks such as causal and temporal relation extraction and discourse parsing.

2 Related Work

Subevent Identification: Only a few studies have focused on identifying subevent relations in text. (Araki et al., 2014) built a logistic regression model to classify the relation between two events into full coreference (FC), subevent parent-child (SP), subevent sister (SS), and no coreference (NC). They improved the prediction of SP relations by performing SS prediction first and using SS prediction results in a voting algorithm. (Glavaš and Šnajder, 2014) trained a logistic regression classifier using a range of lexical and syntactic features and then used Integer Linear Programming (ILP) to enforce document-level coherence for constructing coherent event hierarchies from news. Recently, (Aldawsari and Finlayson, 2019) outperformed previous models for subevent relation prediction using a linear SVM classifier, by introducing several new discourse features and narrative features.

Subevent Knowledge Acquisition: Considering the generalizability issue of supervised contextual classifiers trained on small annotated data, our pilot research on subevent knowledge acquisition (Badgett and Huang, 2016) relies on heuristics, where we first identify sentences in news articles that are likely to contain subevents by exploiting a sentential pattern[3], and then, we extract subevent phrases from those sentences using a phrasal pattern[4]. In addition, this pilot work does not aim to acquire the parent event together with subevents, instead,

[2]While subevents are also spatially contained by the parent event, we did not use this observation to identify candidate subevent relations because the spatial *contained_by* relation between two events is not frequently stated in text.

[3]Subevents often appear in sentences that start or end with characteristic phrases such as "media reports" and "witness said".

[4]Subevent phrases often occur together in conjunction constructions as a sequence of subevent phrases.

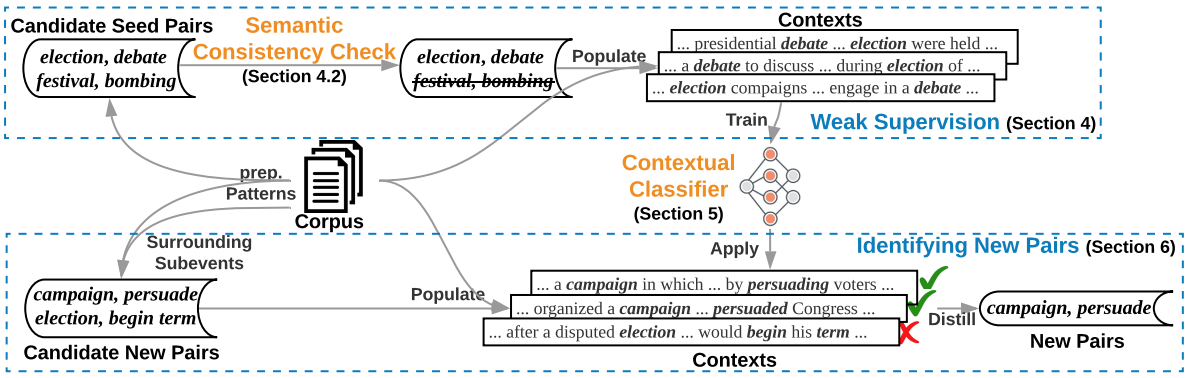

Figure 1: Overview of the Subevent Knowledge Acquisition System

it learns a list of subevent phrases from documents that are known to describe a certain type of event. Specifically, in this work, we only acquired 610 subevent phrases for one type of parent event, civil unrest events. The recent work (Bosselut et al., 2019; Sap et al., 2019) uses generative language models to generate subevent knowledge among many other types of commonsense knowledge.

We can potentially incorporate our learned subevent knowledge into a general event ontology to enrich subevent links in the ontology. For instance, the Rich Event Ontology (REO) (Brown et al., 2017) unifies two existing knowledge resources (i.e., FrameNet (Fillmore et al., 2003) and VerbNet (Kipper et al., 2008)) and two event annotated datasets (i.e., ACE (Doddington et al., 2004) and ERE (Song et al., 2015)) to allow users to query multiple linguistic resources and combine event annotations. However, REO contains few subevent relation links between events.

Identification and Acquisition of other Event Relations: Compared to relatively little research devoted to subevent identification and acquisition, significantly more research has been done for identifying and extracting several other types of event relations, especially temporal relations (Pustejovsky et al., 2003; Chklovski and Pantel, 2004; Bethard, 2013; Llorens et al., 2010; D'Souza and Ng, 2013; Chambers et al., 2014) and causal relations (Girju, 2003; Bethard and Martin, 2008; Riaz and Girju, 2010; Do et al., 2011; Riaz and Girju, 2013; Mirza and Tonelli, 2014, 2016).

3 Overview of the Weakly Supervised Approach

Figure 1 shows the overview of the weakly supervised learning approach for subevent knowledge acquisition. The key of this approach is to identify seed event pairs that are likely to be of the subevent relation in a two-step procedure (Section 4). We first use several temporal relation patterns (e.g., e_i *during* e_j) to identify candidate seed pairs since a child event is usually temporally contained by its parent event; and then, we conduct a definition-guided semantic consistency check to remove spurious subevent pairs that are semantically incompatible and are unlikely to have the subevent relation, e.g., (*festival, bombing*).

Next, we find occurrences of seed pairs in a large text corpus to quickly generate many subevent relation instances, we will also create negative instances to train the subevent relation classifier (Section 5). Then, the trained contextual classifier will be used to identify new event pairs of the subevent relation by examining multiple occurrences of an event pair in text (Section 6). We use the English Gigaword (Napoles et al., 2012) as the text corpus.

4 Weak Supervision

4.1 Seed Event Pair Identification

We use six preposition patterns (i.e., *during, in, amid, throughout, including*, and *within*) to extract candidate seed event pairs. Specifically, we use dependency relations[5] to recognize preposition patterns, and extract the governor word and dependent word of each pattern. We then check whether both words are event triggering words, and try to attach an argument to an event word to form an event phrase that tends to be more expressive and self-contained than a single event word, e.g., *sign agreement* vs *sign*, or, *attack on troops* vs *attack*. We consider both verb event phrases and

[5]We use Stanford dependency relations (Manning et al., 2014), e.g., *prep_during*.

noun event phrases (Appendix A provides more details). We further require that at least one argument is included in an event pair which may be attached to the first or the second event. In other words, we do not consider event pairs in which neither event has an argument.

To select seed subevent pairs, we consider event pairs that co-occur with at least two different patterns for at least three times. In this way, we identified around 43K candidate seed pairs from the Gigaword corpus. However, many candidate seed pairs identified by the preposition patterns only have the temporal *contained_by* relation but do not have the subevent relation. In order to remove such spurious subevent pairs, we present an event definition guided approach next to conduct semantic consistency check between the parent event and the child event of a candidate subevent relation tuple.

4.2 Definition-Guided Semantic Check

The intuition is that the definition of a parent event word describes important aspects of the event's meanings and signifies its potential subevents. For example, based on the definition of *festival*, events related to *"celebrations"*, such as *ceremony being held* and *set off fireworks*, are likely to be correct subevents of *festival*; however, *bomb explosion* and *people being killed* may be distinct events that only happen temporally in parallel with *festival*.

Specifically, we perform semantic consistency checks collectively for many candidate event pairs by considering similarities between events and similarities between the definition of an event and its subevents, and we cluster event phrases into groups so that any two event phrases within a group are semantically compatible. Therefore, when the clustering operation is completed, we will recognize an event pair as a spurious subevent relation pair if its two events fall into different clusters. Next, we describe details on graph construction and the clustering algorithm we used.

4.2.1 Graph Construction

Given a set of event pairs needing the semantic consistency check, we construct an undirected graph $G(V, E)$, where each node in V represents a unique event phrase. We connect event phrases with two types of weighted edges. First, for each candidate subevent relation tuple, we create an edge of weight 1.0 between the parent event and the child event. Second, we create an edge between any two event phrases if their similarity is greater than a certain threshold, and the edge weight is their similarity score. To calculate the similarity between two event phrases, we pair each word from one event phrase (either the event word or an argument) with each word from the other event phrase and calculate similarity between two word embeddings[6], then the similarity between two event phrases is the average of their word pair similarities. We set the similarity threshold as 0.3, after inspecting 200 randomly selected event pairs with their similarities. If two event phrases are already connected because they are a candidate subevent relation pair, we add their similarity score to the edge weight.

Next, we incorporate event definitions by adding new nodes and new edges to the graph. Specifically, for each event phrase that appears as the parent event in some candidate subevent relation tuples, we create a new node for its event word representing the event word definition. If the event word has multiple meanings and therefore multiple definitions, we consider at most five definitions retrieved from WordNet (Miller, 1995) and create one node for each definition, assuming each definition of the parent event will attract different types of children events. Then, we connect each definition node of a parent event with its children events, if their similarity is over the same similarity threshold used previously. The similarity between a definition node and a child event is calculated by exhaustively pairing each non-stop word from the definition sentence and each word from the child event phrase and taking the average of word pair similarities.

4.2.2 The Clustering Algorithm

We use a graph propagation algorithm called Speaker-Listener Label Propagation Algorithm (SLPA) (Xie et al., 2011). SLPA has been shown effective for detecting overlapping clusters (Xie et al., 2013), which is preferred because multiple types of events may share common subevents. For instance, *people being injured* is a commonly seen subevent of conflict events (e.g., *combat*) as well as disaster events (e.g., *hurricane*). In addition, SLPA is self-adapted and can converge to the optimal number of clusters, with no pre-defined number of clusters needed. Event clusters often become stable soon after 50 iterations, to ensure convergence, we ran the algorithm for 60 iterations.

After performing the semantic consistency

[6]We used word2vec word embeddings.

check, we retained around 30K seed event pairs. We find occurrences of these event pairs in the Gigaword corpus and obtained around 388K[7] sentences containing an event pair. These sentences will be used as positive instances to train the contextual classifier.

5 The Contextual Classifier Using BERT

Recently, BERT (Devlin et al., 2019) pretrained on massive data has achieved high performance on various NLP tasks. We fine-tune a pretrained BERT model to build the contextual classifier for subevent relation identification.

BERT model is essentially a bi-directional Transformer-based encoder that consists of multiple layers where each layer has multiple attention heads. Formally, given a sentence with N tokens, each attention head transforms a token vector t_i into query, key, and value vectors q_i, k_i, v_i through three linear transformers. Next, for each token, the head calculates the self-attention scores for all other tokens of the input sentence against this token as the softmax-normalized dot products between two query and key vectors. The output o_i of each attention head is a weighted sum of all value vectors:

$$o_i = \sum_{j=1}^{N} w_{ij} v_j,\ w_{ij} = \frac{\exp(q_i^T k_j)}{\sum_{l=1}^{N} \exp(q_i^T k_l)}$$

In this way, we can obtain N contextualized embeddings $\{o_i\}_{i=1}^N$ for all words $\{w_i\}_{i=1}^N$ in a sentence using the BERT model. To enforce the BERT encoder to look at context information other than the two event trigger words of a subevent pair, e.g., *war, person battle*, we replace the two event trigger words in a sentence with a special token `[MASK]` as the original BERT model did for masking. The contextualized embeddings at two event triggers' positions (two `[MASK]`'s positions) are concatenated and then fed into a feed-forward neural network with a softmax prediction layer for three-way classification, i.e., to predict two subevent relations (parent-child and child-parent relations depending on the textual order of two events) and no subevent relation (Other).

In our experiments, we use the pretrained $\text{BERT}_{\text{base}}$ model provided by (Devlin et al., 2019) with 12 transformer block layers, 768 hidden size and 12 self-attention heads[8]. We train the classifier using cross-entropy loss and Adam (Kingma and Ba, 2015) optimizer with initial learning rate 1e-5, 0.5 dropout, batch size 16 and 3 training epochs.

5.1 Negative Training Instances

High-quality negative training instances that can effectively compete with positive instances are important to enable the classifier to distinguish subevent relations from non-subevent relations. We include two types of negative instances to fine-tune the BERT classifier.

First, we randomly sample sentences that contain an event pair different from any seed pair or candidate pair (Section 6.1) as negative instances. We sample such negative sentences equal to five times of positive sentences, considering that most sentences in a corpus do not contain a subevent relation. Second, we observe that the subevent relation is often confused with temporal and causal event relations because a subevent is strictly temporally contained by its parent event. Therefore, to improve the discrimination capability of the classifier, we also include sentences containing temporally or causally related events as negative instances. Specifically, we apply a similar strategy - using patterns[9] to extract temporal and causal event pairs and then search for these pairs in the text corpus to collect sentences that contain a temporal or causal event pair. Event pairs that co-occur with temporal or causal patterns for at least three times are selected for population. We collected 63K temporally related event pairs and 61K causally related event pairs, which were used to identify 371K sentences that contain one of the event pairs. In total, we obtained around 1.8 million negative training instances.

6 Identifying New Subevent Pairs

We next apply the contextual BERT classifier to identify new event pairs that express the subevent relation. It is unnecessary to test on all possible pairs of events since two random events that co-occur in a sentence have a small chance to have the subevent relation. In order to narrow down the search space, we first identify candidate event pairs that are likely to have the subevent relation.

[7] Some event pairs appear very frequently in the corpus, to encourage diversity of the training data, we keep at most 20 sentences that contain the same event pair.

[8] Our implementation was based on https://github.com/huggingface/transformers.

[9] Three temporal patterns - "following", "before", "after" and seven causal patterns - "lead to", "result in", "result from", "cause", "cause by", "due to", "because of" are used.

Then, we apply the contextual classifier to examine instances of each candidate event pair in order to determine valid subevent relation pairs.

6.1 Candidate Event Pairs

We consider two types of candidate event pairs. First, the preposition patterns used to identify seed subevent relation tuples are again used to identify candidate event pairs, but with less strict conditions. Specifically, we consider event pairs that co-occur with any pattern for at least two times as candidate event pairs. In this way, we identified 1.4 million candidate event pairs from the Gigaword corpus.

Second, when a subevent relation tuple appears in a sentence, it is common to observe other subevents of the same parent event in the surrounding context. Therefore, we collect sentences that contain a seed subevent relation tuple, and identify additional subevents of the same parent event in the two preceding and two following sentences. Furthermore, we observe that the additional subevents often share the subject or direct object with the subevent of the seed tuple, as a consequence, we only consider such event phrases found in the surrounding sentences and pair them with the parent event of the seed tuple to create new candidate event pairs. Using this method, we extracted around 89K candidate event pairs from the Gigaword corpus.

6.2 New Subevent Pair Selection Criteria

We identify a candidate event pair as a new subevent relation pair only if the majority of its sentential contexts, specifically more than 50% of them, were consistently labeled as showing the subevent relation by the BERT classifier. In addition, we disregard rare event pairs and require that at least three instances of an event pair have been labeled as showing the subevent relation.

The full weakly supervised learning process acquires 239K subevent relation pairs, including 30K seed pairs and 209K classifier identified pairs. The subevent knowledge base has 10,318 unique events shown as parent events, and each parent event is associated with 20.1 children events on average.

6.3 An Example Subevent Knowledge Graph

The initial exploration of the learned subevent knowledge shows two interesting observations of event hierarchies. Figure 2 shows an example event graph. First, we can draw a partition of the event space at multiple granularity levels by grouping events based on subevents they share, e.g., the upper and the lower sections of the example event graph illustrate two broad event clusters sharing no subevent, and within each cluster, we see smaller event groups (colored) that share subevents extensively within a group while sharing fewer subevents between groups. Second, subevents encode event semantics and reveal different development stages of the parent events, e.g., subevents of natural disaster events (top left corner) reflect disaster *response* and *recovery* stages.

Seed Pairs	P/R/F1
Before Semantic check	44.9/25.3/32.4
After Semantic check	**55.9/26.2/35.7**

Table 1: Performance of the Contextual Classifier.

7 Intrinsic Evaluation

7.1 Precision of the Contextual Classifier

The contextual classifier is a key component of our learning approach. We evaluate the performance of the BERT contextual classifier on identifying subevent relations against several other types of event-event relations (e.g., temporal, causal relations, etc.), using the Richer Event Description (RED) corpus (O'Gorman et al., 2016) that is comprehensively annotated with rich event-event relations. Since the contextual classifier mainly performs at the sentence level, we only consider to identify intra-sentence subevent relations in the RED dataset[10].

Table 1 shows the comparisons between two training settings - the BERT classifier trained on seed pairs before vs after applying the semantic check (43k vs 30k seed pairs) and their identified training instances. Conducting the semantic check improves the precision of the trained classifier by 11% with no loss on recall. Without using any annotated data, the classifier achieves the precision of 55.9%. While the precision on predicting each sentential context is not perfect, note that we retain a candidate subevent relation pair only if the majority and more than three of its sentential contexts show the subevent relation.

7.2 Accuracy of Acquired Subevent Pairs

We randomly sampled around 1% of acquired subevent pairs, including 300 from seed subevent pairs and 2,090 from newly learned subevent pairs,

[10]RED has 2635 intra-sentence event relations, 530 of them are subevent relations.

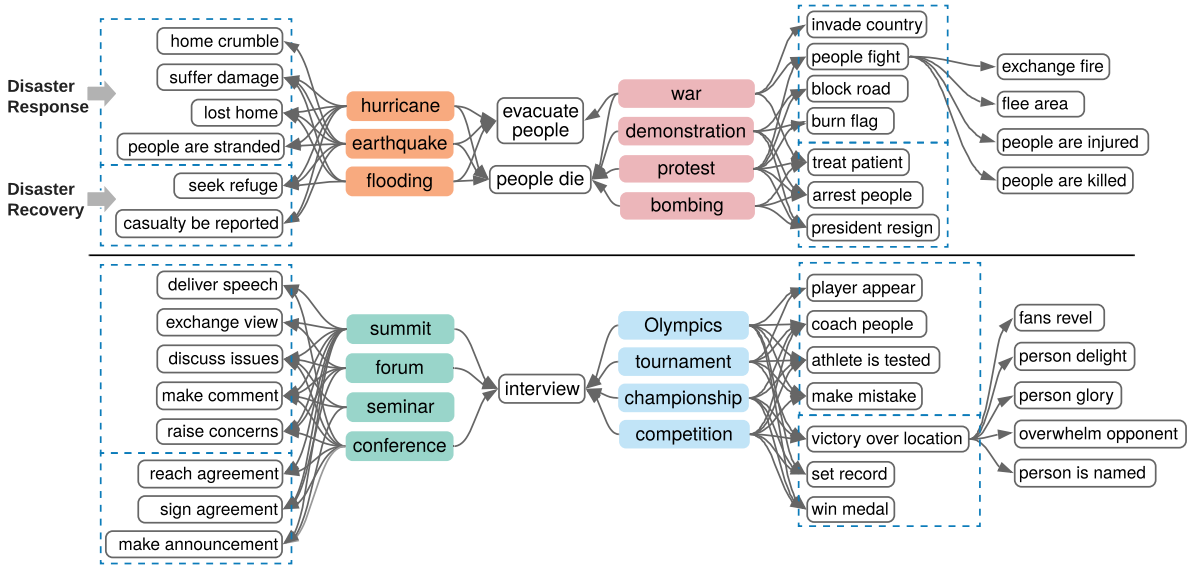

Figure 2: Example Subevent Knowledge Graph (→ denotes the Parent→Child subevent relation).

Method	RED	HiEve
Train and test on intra-sentence event pairs		
Basic BERT Classifier	61.8/52.3/56.6	49.0/46.7/47.9
+ Subevent links	64.8/55.1/59.5	52.5/49.2/**50.8**
+ Event embeddings	67.4/54.2/**60.0**	52.8/46.3/49.4
Train and test on cross-sentence event pairs		
Basic BERT Classifier	65.0/64.8/64.9	33.8/37.4/35.5
+ Subevent links	69.6/66.3/**67.9**	34.0/37.9/35.8
+ Event embeddings	69.2/62.9/65.9	32.5/40.8/**36.2**

Table 2: Subevent Relation Identification. P/R/F1 (%). We predict Parent-Child and Child-Parent subevent relations and report the micro-average performance.

and asked two human annotators to judge whether the subevent relation exists between two events. The two annotators labeled 250 event pairs in common and agreed on 93.6% (234) of them, and the remaining subevent pairs were evenly split between the two annotators. According to human annotations, the accuracy of seed pairs is 91.6% and the accuracy of newly learned event pairs is 89.9%, with the overall accuracy of 90.1%.

7.3 Coverage of Acquired Subevent Pairs

To see whether the acquired subevent knowledge has good coverage of diverse event types, we compare the unique events appearing in the acquired subevent relation tuples with events annotated in two datasets, ACE (Doddington et al., 2004) and KBP (Ellis et al., 2015), both with rich event types annotated and being commonly used for event extraction evaluation. We found that 73.8% (656/889) of events in ACE and 66.9% (934/1396) of events in KBP match with events in the acquired subevent pairs. Because we aim to evaluate the coverage on general event types instead of specific events, we ignore event arguments and only match event word lemmas.

In addition, we compare our learned 239K subevent pairs with the 30K ConceptNet subevent pairs. Interestingly, the two sets only have 311 event pairs in common, which shows that our learning approach extracts subevent pairs from real texts that are often hard to think of by crowd sourcing workers, the approach used by ConceptNet.

8 Extrinsic Evaluation

8.1 Subevent Relation Identification

To find out whether the learned subevent knowledge can be used to improve subevent relation identification in text, we conducted experiments on two datasets, RED[11] and HiEve[12] (Glavaš et al., 2014). In our experiments, we consider intra-sentence and cross-sentence event pairs separately. We randomly split data into five folds and conduct cross-validation for evaluation. We fine-tune the same BERT model using RED or HiEve annotations to

[11]RED has 530 intra-sentence and 415 cross-sentence subevent relations.

[12]HiEve annotated 3,200 event mentions and their subevents as well as coreference relations in 100 documents. We first extended the subevent annotations using transitive closure rules and coreference relations (Glavaš et al., 2014; Aldawsari and Finlayson, 2019), which produces 490 intra-sentence and 3.1K cross-sentence subevent relations.

Method	Macro	Acc	Comparison	Contingency	Expansion	Temporal
Base Model	50.8/47.8/49.0	56.42	43.8/39.0/41.3	44.7/51.3/47.8	66.6/65.7/66.2	48.2/35.0/40.6
+ Subevent (ours)	53.2/49.5/**51.0**	**59.08**	44.3/34.9/39.1	49.2/46.1/47.6	66.3/73.3/69.6	52.8/43.8/47.9

Table 3: Multi-class Classification Results on the PDTB dataset. We report accuracy (Acc), macro-average (Macro) P/R/F1 (%) over four implicit discourse relation categories as well as performance on each category.

Method	RED	TimeBank
Train and test on intra-sentence event pairs		
Basic BERT Classifier	59.9/68.2/63.8	66.8/62.2/64.4
+ Subevent links	61.3/69.1/**65.0**	65.4/67.0/**66.2**
+ Event embeddings	59.8/69.8/64.4	64.1/68.1/66.1
Train and test on cross-sentence event pairs		
Basic BERT Classifier	38.4/37.4/37.9	44.1/48.4/**46.1**
+ Subevent links	51.8/40.7/45.5	45.3/40.7/42.8
+ Event embeddings	52.4/42.3/**46.8**	43.5/47.6/45.4

Table 4: Temporal Relation Identification. P/R/F1 (%). We predict Before and After temporal relations and report the micro-average performance.

Method	RED	ESC
Train and test on intra-sentence event pairs		
Basic BERT Classifier	64.7/62.6/63.6	44.9/52.2/48.3
+ Subevent links	64.1/66.5/65.3	44.9/54.5/49.2
+ Event embeddings	65.2/66.8/**66.0**	45.9/53.4/**49.4**
Train and test on cross-sentence event pairs		
Basic BERT Classifier	20.0/14.3/16.7	30.3/23.9/26.7
+ Subevent links	28.4/26.1/**27.2**	34.0/22.7/27.2
+ Event embeddings	28.0/25.2/26.6	32.1/25.4/**28.4**

Table 5: Causal Relation Identification. P/R/F1 (%). We predict Cause-Effect and Effect-Cause relations and report the micro-average performance.

predict subevent relations vs others[13][14]. Note that for cross-sentence event pairs, we simply concatenate two sentences and insert in between the special token `[SEP]` used in the original BERT.

We propose two methods to incorporate the learned subevent knowledge. **1) Subevent links**. For a pair of events to classify in the RED or HiEve dataset, we check if they match with our learned subevent relation tuples. We ignore event arguments for matching events and only consider to match event word lemmas, for this reason, one pair of events might match with multiple learned subevent relations. We count subevent relations that match with a given event pair, (X, Y), in two directions (X $\overset{subevent}{\rightarrow}$ Y) and (Y $\overset{subevent}{\rightarrow}$ X) separately, and encode the log values of the two counts in a vector. **2) Event embedding**. Subevent relations can be used to build meaningful event embeddings to have the embeddings of a parent event and a child event preserve the subevent relation between them. Therefore, we train a BiLSTM encoder[15] to build event phrase embeddings, using the knowledge representation learning model TransE (Bordes et al., 2013)[16] such that $\mathbf{p} + \mathbf{r} \approx \mathbf{c}$ given a parent-child event pair (p, c) having the subevent relation r. We will use the trained BiLSTM encoder to obtain an embedding for an event phrase in the RED or HiEve dataset.

Finally, for subevent relation identification, we concatenate two event word representations obtained by the BERT encoder with either a subevent link vector or two event embeddings obtained using the above two methods. Results are shown in Table 2. We can see that compared to the basic BERT classifier, incorporating learned subevent knowledge achieves better performance on both datasets, for both intra-sentence and cross-sentence cases.

8.2 Temporal and Causal Relation Identification

Subevents indicate how an event emerges and develops, and therefore, the learned subevent knowledge can further be used to identify other semantic relations between events, such as temporal and causal relations. For evaluation, we use the same RED [17] dataset plus two more datasets, TimeBank v1.2[18] (Pustejovsky et al., 2003) and Event Storyline Corpus (ESC) v1.5[19] (Caselli and Inel, 2018),

[13]For the RED dataset, we consider all the annotated event-event relations in RED other than subevent relations as others.

[14]For the HiEve dataset, we exhaustively create event mention pairs among all the annotated event mentions in HiEve and consider all the mention pairs that were not annotated with the subevent relation as others. In this way, we generated 3.5K intra-sentence and 59.5K cross-sentence event mention pairs as others.

[15]The BiLSTM has the hidden size of 50 and uses max-pooling to encode an event phrase.

[16]We trained TransE for 20 iterations.

[17]RED has 1104 (1010) intra-sentence and 182 (119) cross-sentence temporal (causal) relations. We consider all the annotated event-event relations in RED other than temporal (causal) relations as others.

[18]TimeBank has 1,122 intra-sentence and 247 cross-sentence "before/after" temporal relations. We consider all the annotated event-event relations in TimeBank other than "before/after" relations as others.

[19]ESC has 1,649 intra-sentence and 3,952 cross-sentence causal relations. We exhaustively create event mention pairs

dedicated to evaluate temporal relation and causal relation identification systems respectively. We use the same experimental settings, including 5-fold cross-validations and evaluating predictions of intra- and cross-sentence cases separately. In addition, we repurpose the BERT model to predict temporal relations vs others or predict causal relations vs others, and we use the same two methods to incorporate the learned subevent knowledge.

Table 4 and 5 show results of temporal and causal relation identification. We can see that subevent knowledge has little impact for identifying intra-sentence temporal and causal relations that may heavily rely on local contextual patterns within a sentence. However, for identifying the more challenging cross-sentence cases that usually have little contextual clues to rely on, the learned subevent knowledge has noticeably improved the system performance on both datasets. This is true for both temporal relations and causal relations. Overall, the systems achieved the best performance when using the event embedding approach to incorporate subevent knowledge.

8.3 Implicit Discourse Relation Classification

We expect subevent knowledge to be useful for classifying discourse relations between two text units in general because subevent descriptions often elaborate and provide a continued discussion of a parent event introduced earlier in text. For experiments, we used our recent discourse parsing system (Dai and Huang, 2019) that easily incorporates external event knowledge as a regularizer into a two-level hierarchical BiLSTM model (Base Model) for paragraph-level discourse parsing. The experimental setting is exactly the same as in (Dai and Huang, 2019).

Table 3 reports the performance of implicit discourse relation classification on PDTB 2.0 (Prasad et al., 2008). Incorporating the acquired subevent pairs (239K) into the Base Model improves the overall macro-average F1-score and accuracy by 2.0 and 2.6 points respectively, which is non-trivial considering the challenges of implicit discourse relation identification. The performance improvements are noticeable on both the expansion relation and the temporal relation categories.

among all the annotated event mentions in ESC and consider all the mention pairs that were not annotated with the causal relation as others. In this way, we generated 4.1K intra-sentence and 34K cross-sentence event mention pairs as others.

9 Conclusions

We have presented a novel weakly supervised learning framework for acquiring subevent knowledge and built the first large scale subevent knowledge base containing 239K subevent tuples. Evaluation showed that the acquired subevent pairs are of high quality (90.1% of accuracy) and cover a wide range of event types. We performed extensive evaluations showing that the harvested subevent knowledge not only improves subevent relation extraction, but also improve a wide range of NLP tasks such as causal and temporal relation extraction and discourse parsing. In the future, we would like to explore uses of the subevent knowledge base for other event-oriented applications such as event tracking.

Acknowledgments

We thank our anonymous reviewers for providing insightful review comments. We gratefully acknowledge support from National Science Foundation (NSF) via the awards IIS-1942918 and IIS-1755943. This work was also supported by DARPA/I2O and U.S. Army Research Office Contract No. W911NF-18-C-0003 under the World Modelers program, and in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the BETTER program. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies, either expressed or implied, of NSF, ODNI, IARPA, the Department of Defense or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Mohammed Aldawsari and Mark Finlayson. 2019. Detecting subevents using discourse and narrative features. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4780–4790.

Jun Araki, Zhengzhong Liu, Eduard H Hovy, and Teruko Mitamura. 2014. Detecting subevent structure for event coreference resolution. In *LREC*, pages 4553–4558.

Allison Badgett and Ruihong Huang. 2016. Extracting subevents via an effective two-phase approach.

In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 906–911.

Steven Bethard. 2013. Cleartk-timeml: A minimalist approach to tempeval 2013. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 2, pages 10–14.

Steven Bethard and James H Martin. 2008. Learning semantic links from a corpus of parallel temporal and causal relations. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 177–180. Association for Computational Linguistics.

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.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for knowledge graph construction. In *Association for Computational Linguistics (ACL)*.

Susan Brown, Claire Bonial, Leo Obrst, and Martha Palmer. 2017. The rich event ontology. In *Proceedings of the Events and Stories in the News Workshop*, pages 87–97, Vancouver, Canada. Association for Computational Linguistics.

Tommaso Caselli and Oana Inel. 2018. Crowdsourcing StoryLines: Harnessing the crowd for causal relation annotation. In *Proceedings of the Workshop Events and Stories in the News 2018*, pages 44–54, Santa Fe, New Mexico, U.S.A. Association for Computational Linguistics.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.

Zeyu Dai and Ruihong Huang. 2019. A regularization approach for incorporating event knowledge and coreference relations into neural discourse parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2974–2985.

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

Quang Xuan Do, Yee Seng Chan, and Dan Roth. 2011. Minimally supervised event causality identification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 294–303. Association for Computational Linguistics.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, page 1. Lisbon.

Jennifer D'Souza and Vincent Ng. 2013. Classifying temporal relations with rich linguistic knowledge. In *HLT-NAACL*, pages 918–927.

Joe Ellis, Jeremy Getman, Dana Fore, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie Strassel. 2015. Overview of linguistic resources for the tac kbp 2015 evaluations: Methodologies and reults. In *Proceedings of the TAC KBP 2015 Workshop*, pages 16–17.

Charles J Fillmore, Christopher R Johnson, and Miriam RL Petruck. 2003. Background to framenet. *International journal of lexicography*, 16(3):235–250.

Roxana Girju. 2003. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering-Volume 12*, pages 76–83. Association for Computational Linguistics.

Goran Glavaš and Jan Šnajder. 2014. Constructing coherent event hierarchies from news stories. In *Proceedings of TextGraphs-9: the workshop on Graph-based Methods for Natural Language Processing*, pages 34–38.

Goran Glavaš, Jan Šnajder, Parisa Kordjamshidi, and Marie-Francine Moens. 2014. Hieve: A corpus for extracting event hierarchies from news stories. In *Proceedings of 9th language resources and evaluation conference*, pages 3678–3683. ELRA.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A large-scale classification of english verbs. *Language Resources and Evaluation*, 42(1):21–40.

Hector Llorens, Estela Saquete, and Borja Navarro. 2010. Tipsem (english and spanish): Evaluating crfs and semantic roles in tempeval-2. In *Proceedings of*

the 5th International Workshop on Semantic Evaluation, pages 284–291. Association for Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 55–60.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Paramita Mirza and Sara Tonelli. 2014. An analysis of causality between events and its relation to temporal information. In *COLING*, pages 2097–2106.

Paramita Mirza and Sara Tonelli. 2016. Catena: Causal and temporal relation extraction from natural language texts. In *The 26th International Conference on Computational Linguistics*, pages 64–75.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 95–100. Association for Computational Linguistics.

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

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

R. Prasad, N. Dinesh, Lee A., E. Miltsakaki, L. Robaldo, Joshi A., and B. Webber. 2008. The Penn Discourse Treebank 2.0. In *lrec2008*.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The timebank corpus. In *Corpus linguistics*, volume 2003, page 40.

Mehwish Riaz and Roxana Girju. 2010. Another look at causality: Discovering scenario-specific contingency relationships with no supervision. In *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*, pages 361–368. IEEE.

Mehwish Riaz and Roxana Girju. 2013. Toward a better understanding of causality between verbal events: Extraction and analysis of the causal power of verb-verb associations. In *Proceedings of the annual SIGdial Meeting on Discourse and Dialogue (SIGDIAL)*. Citeseer.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.

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

Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. 2013. Overlapping community detection in networks: The state-of-the-art and comparative study. *Acm computing surveys (csur)*, 45(4):43.

Jierui Xie, Boleslaw K Szymanski, and Xiaoming Liu. 2011. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 344–349. IEEE.

A Event Representations

We consider both verb event phrases and noun event phrases.

Verb Event Phrases: To ensure good coverage of regular event pairs, we consider all verbs[20] as event words except possession verbs[21]. The thematic patient of a verb refers to the object being acted upon and is essentially part of an event, therefore, we first consider the patient of a verb in forming an event phrase[22]. The agent is also useful to specify an event especially for an intransitive verb event, which does not have a patient. Therefore, we include the agent of a verb event in an event phrase if its patient was not found. The patient or agent of a verb is identified using dependency relations[23]. If neither a patient nor an agent was found, we include a preposition phrase (a preposition and its object) that modifies a verb in the event representation to form an event phrase. Example verb event phrases are "agreement be *signed*" and "*occupy* territory".

Noun Event Phrases: We include a preposition phrase (a preposition and its object)that modifies a noun event in the event representation to form a noun event phrase. We first consider a preposition phrase headed by the preposition *of*, then a preposition phrase headed by the preposition *by*, lastly a preposition phrase headed by any other preposition. Example noun event phrases are "*ceremony* at location" and "*attack* on troops".

Note that many noun words do not refer to an event, therefore, we apply two strategies to quickly compile a list of noun event words. First, we obtain a list of deverbal nouns[24] (5028 event nouns) by querying each noun in WordNet (Miller, 1995) and checking if its root word form has a verb sense. Second, we identify five intuitive textual patterns, e.g., *participate in* EVENT, and extract their prepositional direct objects as potential noun events. The five patterns are: *participate in* EVENT, *involve in* EVENT, *engage in* EVENT, *play role in* EVENT and *series of* EVENT. We rank extractions first by the number of times they occur with these patterns and then by the number of unique patterns they occur with. We next quickly went through the top 5,000 nouns and manually removed non-event words, which results in 3154 noun event words.

Event Phrase Generalization: Including arguments into event representations generates event phrases that are too specific though. In order to obtain generalized event phrase forms, we replace named entity arguments with their entity types (Manning et al., 2014). We also replace personal pronouns with the entity type PERSON.

[20] We used POS tags to detect verb events.

[21] We determined that possession verbs, such as "own", "have" and "contain", mainly express the ownership status so we discarded these event phrases.

[22] In particular, we require a light verb (e.g., do, make, take etc.) to have a direct object because light verbs have little semantic content of their own.

[23] We use Stanford dependency relations (Manning et al., 2014). We identify the patient as the direct object of an active verb or the subject of a passive verb; we identify the agent as the subject of an active verb or the object of preposition *by* modifying a passive verb.

[24] Derivative nouns ending with suffixes -er, -or are discarded.