WEAKLY-SUPERVISED LEARNING APPROACHES FOR EVENT KNOWLEDGE ACQUISITION AND EVENT DETECTION

A Dissertation

by

WENLIN YAO

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Ruihong Huang
Committee Members,	James Caverlee
	Yoonsuck Choe
	Laura Mandell
Head of Department,	Scott Schaefer

December 2020

Major Subject: Computer Science

ABSTRACT

Capabilities of detecting events and recognizing temporal, subevent, or causal relations among events can facilitate many applications in natural language understanding. However, supervised learning approaches that previous research mainly uses have two problems. First, due to the limited size of annotated data, supervised systems cannot sufficiently capture diverse contexts to distill universal event knowledge. Second, under certain application circumstances such as event recognition during emergent natural disasters, it is infeasible to spend days or weeks to annotate enough data to train a system. My research aims to use weakly-supervised learning to address these problems and to achieve automatic event knowledge acquisition and event recognition.

In this dissertation, I first introduce three weakly-supervised learning approaches that have been shown effective in acquiring event relational knowledge. Firstly, I explore the observation that regular event pairs show a consistent temporal relation despite of their various contexts, and these rich contexts can be used to train a contextual temporal relation classifier to further recognize new temporal relation knowledge. Secondly, inspired by the double temporality characteristic of narrative texts, I propose a weakly supervised approach that identifies 287k narrative paragraphs using narratology principles and then extract rich temporal event knowledge from identified narratives. Lastly, I develop a subevent knowledge acquisition approach 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 guide the identification of subevents. I collect rich weak supervision to train a contextual BERT classifier and apply the classifier to identify new subevent knowledge.

Recognizing texts that describe specific categories of events is also challenging due to language ambiguity and diverse descriptions of events. So I also propose a novel method to rapidly build a fine-grained event recognition system on social media texts for disaster management. My method creates high-quality weak supervision based on clustering-assisted word sense disambiguation and enriches tweet message representations using preceding context tweets and reply tweets in building event recognition classifiers.

ACKNOWLEDGMENTS

First and foremost I want to give special thanks to Dr. Ruihong Huang for the support and encouragement I received during my Ph.D. journey. I am also grateful to all committee members Dr. James Caverlee, Dr. Yoonsuck Choe and Dr. Laura Mandell.

I would also like to acknowledge other TAMU NLP group members, especially Prafulla Choubey, Zeyu Dai and Pei Chen for their insightful feedback and valuable help on my research projects.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Dr. Ruihong Huang (advisor), Dr. James Caverlee, and Dr. Yoonsuck Choe of the Department of Computer Science and Engineering and Dr. Laura Mandell of the Department of English.

All work for the dissertation was completed by the student, in collaboration with Dr.Ruihong Huang (advisor) of the Department of Computer Science and Engineering.

Funding Sources

This work was made possible in part by National Science Foundation (NSF) via the awards IIS-1759537, IIS-1755943 and IIS-1942918. 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.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

Events describe the interactions between agents (e.g., people, organizations, countries, etc.) and objects involved. Understanding text-internal events is a critical step to interpret complicated dynamics of the world evolving every day. It is feasible only when we can accurately detect particular event types and comprehensively recognize how different events are interconnected through relational links. Therefore, developing computational models to recognize specified events and their relations (e.g., temporal, subevent and causal relations) is a long-held goal in Natural Language Processing (NLP) research. For example, automatic recognizing relations among events helps us to model the world, which supports event prediction, crisis forecast and risk management. Automatic detecting life-threatening events during natural disasters provides us real-time event sensing, which can help local authorities and responders to facilitate evacuation operations or finding victims in need of help. However, both event relation identification and event recognition are very challenging.

First, recognizing the relation between two events is very challenging as the relation can be described in dramatically different contexts depending on domains and pairs of events, signifying different semantic meanings. In order to capture various contexts, large amounts of labeled data are needed to train a high-coverage relation classifier. However, almost all existing datasets that contain event-event relation annotations are limited in size and context diversity, such as Automatic Context Extraction (ACE) [Strassel et al., 2008] and TimeBank [Pustejovsky et al., 2003], HiEve [Glavaš et al., 2014] which generally contain one to two hundred documents. Most of the existing relation classifiers are trained using these small manually annotated datasets, relying on sophisticated lexical, grammatical, linguistic (e.g., tenses and aspects of events), semantic (e.g., semantic roles and lexicon derived features) and discourse (e.g., discourse connectives [Mirza and Tonelli, 2014b]) features.

Luckily, certain events often co-occur in a particular event relation, which can be extracted as general event ontology knowledge. For example, people often go to *work* after *graduation* with a

degree, people often *order*, *eat* and *pay* when *dining* in a restaurant, *hurricanes* often cause *flooding* that may further cause *houses collapse*. Such event "before/after" temporal knowledge, "subevent-parentevent" knowledge and "cause-effect" knowledge can be extracted as general event ontology knowledge. Event ontology knowledge reveals relations between events despite of their complex contexts, which facilitates comprehensive reasoning in documents thus benefits various NLP applications, including event tracking, event timeline generation, text summarization and question answering. While being in high demand, event ontology knowledge is lacking and difficult to obtain. Existing knowledge bases, such as Freebase [Bollacker et al., 2008] or Probase [Wu et al., 2012], often contain rich knowledge about entities, e.g., the birthplace of a person, but contain little event knowledge. Several approaches have been proposed to acquire event knowledge from a text corpus, by either crowdsourcing knowledge from human [Baker et al., 1998, Havasi et al., 2007, Rashkin et al., 2018], utilizing textual patterns [Chklovski and Pantel, 2004b] or applying pretrained discourse parser [Zhang et al., 2020].

Second, recognizing texts that describe specified event categories is also challenging due to language ambiguity and diverse expressions of events. In order to recognize life-threatening emergency events on social media for reducing life and property losses during natural disasters, previous research focuses on supervised classification approaches or burst topics detection. However, it is critical to build this event recognizer rapidly during natural disasters but supervised approaches require a carefully labeled dataset that is collected by asking human annotators to label a large number of data instances one by one. Such data labeling process usually takes days or weeks, which makes supervised learning schema not suitable during natural disasters. While burst topics detection may reliably identify a disaster is happening but cannot recognize fine-grained types of life-threatening events (e.g., mandatory evacuations, road closures, etc.).

In this dissertation, I propose several weakly-supervised learning approaches to address the above challenges. Specifically, I have successfully applied three weakly-supervised learning approaches to acquire event temporal knowledge and subevent knowledge. I also have developed a weakly-supervised event recognition system to recognize fine-grained event categories on social

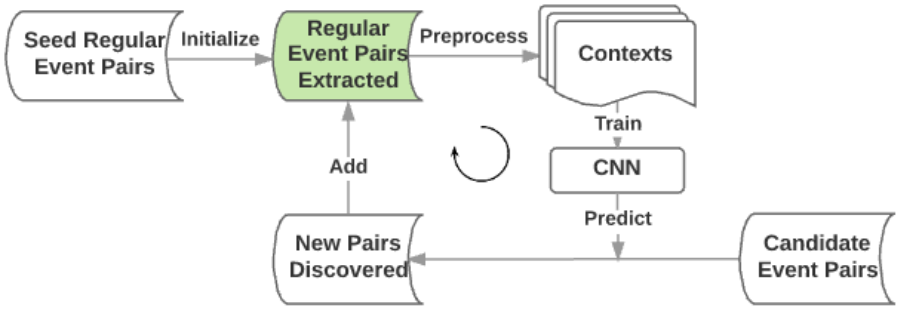

Figure 1.1: Overview of the Bootstrapping System for Temporal Knowledge Acquisition. Reprinted with permission from Yao et al. [2017].

media texts during natural disasters. My approaches can effectively collect weak supervision automatically from texts and has been shown effective in acquiring event relational knowledge and detecting fine-grained event types on social media texts.

1.1 Weakly-supervised Learning for Acquiring Event Knowledge

1.1.1 Temporal Knowledge Acquisition via CNN Contextual Classifiers

First, I observed that event pairs presenting regularities tend to show the same temporal relation despite of various contexts they may occur in. For instance, *arrest* events tend to happen after *attack* events, and the following sentential contexts all indicate the same temporal relation:

Under pressure following suicide attacks, police arrested scores of activists on Monday.

Two men were arrested on suspicion of carrying out the Mumbai attacks.

Carlos was arrested in Sudan in August in connection with two bomb attacks in France in 1982.

Mamdouh Habib was arrested in Pakistan three weeks after the Sept.11 attacks.

Leveraging this key observation, I propose a bootstrapping approach (Figure 1.1) that focuses on recognizing *after* or *before* temporal relations and substantially reduces the reliance on human annotated data. I start by identifying regular event pairs that have occurred enough times with an explicit temporal pattern, i.e., EV_A *after (before)* EV_B. I then populate these seed event pairs in a large unlabeled corpus to quickly collect hundreds of thousands of sentences that contain a regular event pair, which are then used as training instances to obtain an initial contextual temporal

Michael Kennedy graduated with a bachelor's degree from Harvard University in 1980. He married his wife, Victoria, in 1981 and attended law school at the University of Virginia. After receiving his law degree, he briefly worked for a private law firm before joining Citizens Energy Corp. He took over management of the corporation, a non-profit firm that delivered heating fuel to the poor, from his brother Joseph in 1988. Kennedy expanded the organization goals and increased fund raising.

Beth paid the taxi driver. She jumped out of the taxi and headed towards the door of her small cottage. She reached into her purse for keys. Beth entered her cottage and got undressed. Beth quickly showered deciding a bath would take too long. She changed into a pair of jeans, a tee shirt, and a sweater. Then, she grabbed her bag and left the cottage.

Figure 1.2: Two narrative examples. Reprinted with permission from Yao and Huang [2018].

relation classifier. Next, the classifier is applied back to the text corpus and label new sentential contexts that indicate a specific *after* or *before* temporal relation between events. Then new regular event pairs can be identified, which are event pairs that have a majority of their sentences labeled as describing a particular temporal relation. The newly identified regular event pairs will be used to augment seed event pairs and identify more temporal relation sentential contexts in the unlabeled corpus. The bootstrapping learning process iterates. Through this weakly supervised learning method, I obtain both a contextual temporal relation classifier and a list of regular event pairs that usually show a particular "after/before" temporal relation.

1.1.2 Temporal Knowledge Acquisition via Identifying Narratives

Second, inspired by the double temporality characteristic of narrative texts, I propose a novel approach for acquiring rich temporal "before/after" event knowledge across sentences via identifying narrative stories. The double temporality states that a narrative story often describes a sequence of events following the chronological order and therefore, the temporal order of events matches with their textual order [Walsh, 2001, Riedl and Young, 2010, Grabes, 2013]. Therefore, we can easily distill temporal event knowledge if we have identified a large collection of narrative texts. Consider the two narrative examples in figure 7.5, where the top one is from a news article of New York Times and the bottom one is from a novel book. From the top one, we can easily extract one chronologically ordered event sequence {graduated, marry, attend, receive, work, take

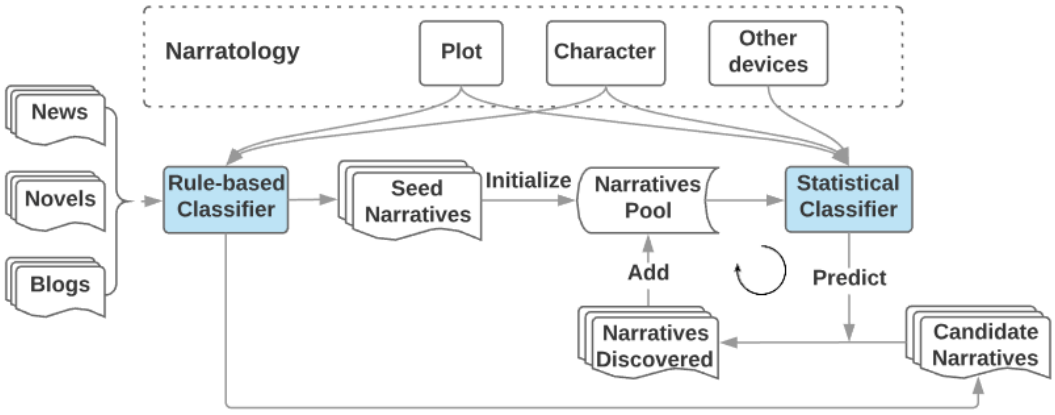

Figure 1.3: Overview of the Narrative Learning System. Reprinted with permission from Yao and Huang [2018].

over, expand, increase}, with all events related to the main character Michael Kennedy. While some parts of the event sequence are specific to this story, the event sequence contains regular event temporal relations, e.g., people often {graduate} first and then get {married}, or {take over} a role first and then {expand} a goal. Similarly, from the bottom one, we can easily extract another event sequence {pay, jump out, head, reach into, enter, undress, shower, change, grab, leave} that contains routine actions when people take a shower and change clothes.

There has been recent research on narrative identification from blogs by building a text classifier in a supervised manner [Gordon and Swanson, 2009, Ceran et al., 2012]. However, narrative texts are common in other genres as well, including news articles and novel books, where little annotated data is readily available. Therefore, in order to identify narrative texts from rich sources, I develop a weakly supervised method (shown in Figure 1.3) that can quickly adapt and identify narrative texts from different genres, by heavily exploring the principles that are used to characterize narrative structures in narratology studies. It is generally agreed in narratology [Forster, 1962, Mani, 2012, Pentland, 1999, Bal, 2009] that a narrative is a discourse presenting a sequence of events arranged in their time order (the plot) and involving specific characters (the characters). First, I derive specific grammatical and entity co-reference rules to identify narrative paragraphs that each contains a sequence of sentences sharing the same actantial syntax structure (i.e., *NP VP*

describing *a character did something*) [Greimas, 1971] and mentioning the same character. Then, I train a classifier using the initially identified seed narrative texts and a collection of grammatical, co-reference and linguistic features that capture the two key principles and other textual devices of narratives. Next, the classifier is applied back to identify new narratives from raw texts. The newly identified narratives will be used to augment seed narratives and the bootstrapping learning process iterates until no enough new narratives can be found.

Then by leveraging the double temporality characteristic of narrative paragraphs, I distill general temporal event knowledge. Specifically, I extract event pairs as well as longer event sequences consisting of strongly associated events that often appear in a particular textual order in narrative paragraphs, by calculating Causal Potential [Beamer and Girju, 2009, Hu et al., 2013] between events.

1.1.3 Subevent Knowledge Acquisition via Fine-tuning a BERT Classifier

The third model focuses on subevent knowledge acquisition. 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 and recognizing the subevent relation between events in a story. 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 its relations with other events (e.g., causal and temporal relations), and therefore has great potential to benefit various event oriented applications such as 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, I 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.

I propose to conduct weakly supervised learning and train a wide-coverage contextual classifier to acquire diverse event pairs of the subevent relation from text. Figure 5.1 provides the overview of my subevent knowledge acquisition system. I start by creating **weak supervision**, where I 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, I 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 indicate the temporal containment relationship[1], and (2) the definition of the parent event is useful to prune spurious subevent tuples away to improve the quality.

Specifically, I 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, I 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, I 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. I train the con-

[1]While subevents are also spatially contained by the parent event, I 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.

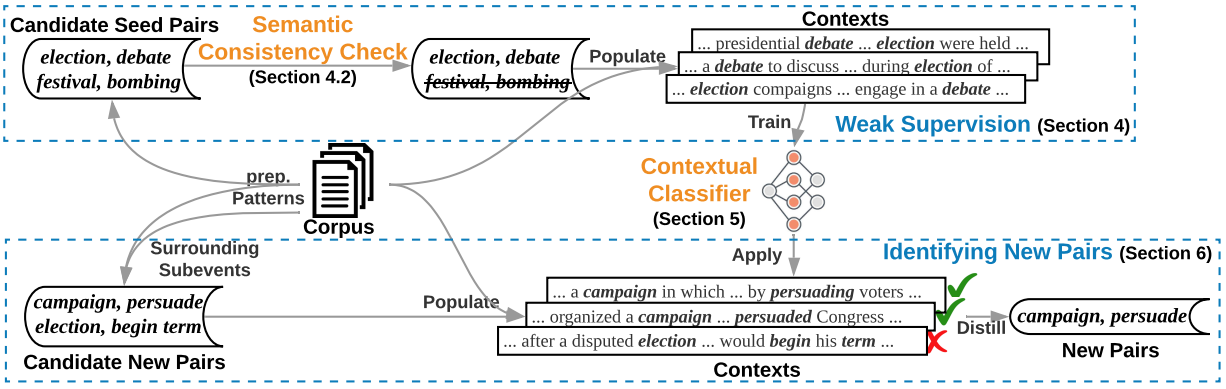

Figure 1.4: Overview of the Subevent Knowledge Acquisition System. Reprinted with permission from Yao et al. [2020a].

textual subevent relation classifier by fine-tuning the pretrained BERT model Devlin et al. [2019]. I then apply the contextual BERT classifier to identify new event pairs that have the subevent relation.

I 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.

1.1.4 Constructing Event Knowledge Graph and Learning Distributed Representations of Events

I have introduced three weakly-supervised approaches for acquiring event temporal knowledge and event subevent knowledge. Besides temporal and subevent relations, causal relations are also important in NLP applications, particularly event prediction and causal reasoning. Recall that my weakly-supervised approach developed for subevent knowledge acquisition only relies on weak supervision, where I use linguistic patterns to identify the initial set of an event relation from a big text corpus. Thus, I replace initial subevent linguistic patterns with causality linguistic patterns to fine-tune a BERT classifier to extract event causal knowledge. Finally, I merge all acquired event knowledge into a general event knowledge graph. The final event knowledge graph contains 126K event nodes, 174K *happens-before* edges, 239K *parent-subevent* edges and 49K *cause-effect* edges. To facilitate downstream NLP usage, I investigate different ways to learn distributed representations of events on constructed event knowledge graph. Experiments show that event knowledge graph embeddings can improve the baseline performance on several benchmark datasets.

1.2 Weakly-supervised Learning for Detecting Events on Social Media

Identifying events of specified types (e.g., life-threatening events) in texts is crucial in event tracking and crisis management. Recently, people increasingly use social media to report emergencies, provide real-time situation updates, offer or seek help or share information during natural disasters. During the devastating hurricane *Harvey* for example, the local authorities and disaster responders as well as the general public had frequently employed Twitter for real-time event sensing, facilitating evacuation operations, or finding victims in need of help.

Considering the large volume of social media messages, it is necessary to achieve automatic recognition of life-threatening events based on individual messages for improving the use of social media during disasters. This task is arguably more challenging than the well-studied collection-based event detection task on social media that often relies on detecting a burst of words over a collection of messages, especially considering the unique challenges of social media texts being

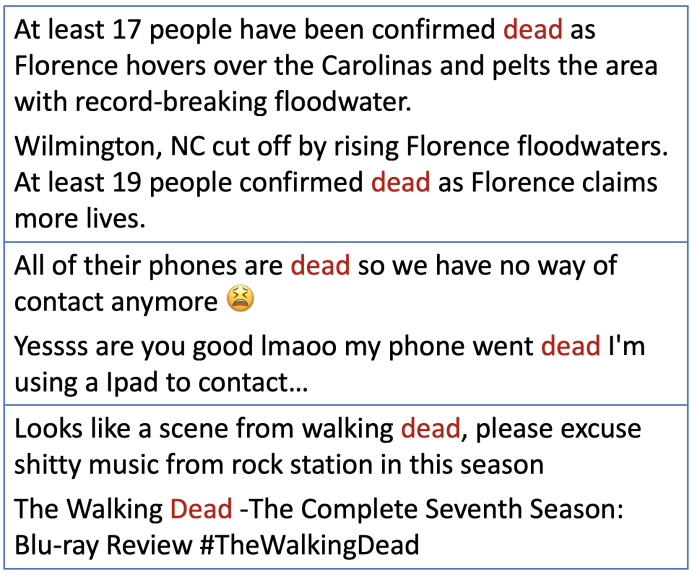

Figure 1.5: Examples of three senses of the word "dead". Reprinted with permission from Yao et al. [2020b].

extremely noisy and short.

To facilitate disaster management, especially during the time-critical disaster *response* phase, it is vital to build event recognizers rapidly. However, the typical supervised learning paradigm requires a carefully labeled dataset that is normally created by asking human annotators to go through a large number of data instances and label them one by one, and the data labeling procedure usually takes days at least. For fast deployment, I propose a novel data labeling method and an overall weakly supervised learning approach that quickly builds reliable fine-grained event recognizers.

Specifically, to quickly label data, I explore the idea of identifying several high-quality event keywords and populating the keywords in a large unlabeled tweet collection. But, I quickly realize that it is essentially impossible to find an event keyword that is not ambiguous and has only one meaning in social media. Taking the word "dead" for example, in addition to the meaning of "losing life", "dead" is also frequently used to refer to phones being out of power or a TV series "walking dead", with example tweets shown in Figure 1.5. It is a challenging problem because

current automatic word sense disambiguation systems only achieve mediocre performance and may not work well on tweets with little in-domain training data, especially considering that many word senses appearing in tweets may even not appear in conventional sense inventories at all, e.g., the word "dead" referring to the TV series "walking dead".

Luckily, I observe that tweets adopting one common sense of an event keyword often share content words and can be easily grouped together. This observation is consistent with previous research on unsupervised word sense disambiguation Yarowsky [1995], Navigli and Lapata [2010]. Therefore, I first cluster keyword identified noisy tweets using an automatic clustering algorithm and rank tweet clusters based on the number of tweets in each cluster. Next, I conduct manual Word Sense Disambiguation (WSD) by simply asking a domain expert to quickly go through the top-ranked clusters and judge whether each tweet cluster show the pertinent meaning of an event keyword, based on an inspection of five example tweets randomly sampled from a cluster. The domain expert is instructed to stop once 20 pertinent clusters have been identified. In this way, I significantly improved the quality of keyword identified tweets, requiring only 1-2 person-hours of manual cluster inspection time. Note that this is the only step in the overall weakly supervised approach that requires human supervision.

Next, I use the rapidly created labeled data to train a recurrent neural net classifier and learn to recognize fine-grained event categories for individual Twitter messages. But tweets are often rather short, and it is difficult to make event predictions solely based on the content of a tweet itself. Instead, I use preceding context tweets posted by the same user as well as replies from other users, together with the target tweet, in a multi-channel neural net to predict the right event category. The observation is that the context tweets as well as reply tweets can both provide essential clues for inferring the topic of the target tweet. For instance, the upper example of Figure 1.6 shows that the two preceding tweets from the same user indicate the third tweet is asking about the location for evacuation; and the lower example shows that based on the reply tweet messages, I can infer the first tweet is regarding water release of reservoir even having no external knowledge about Addicks/Barker.

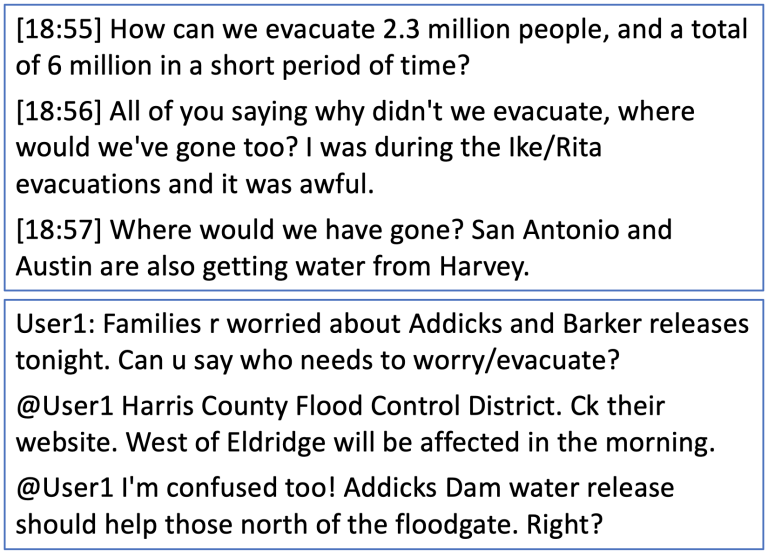

Figure 1.6: Examples with context and reply tweets. Reprinted with permission from Yao et al. [2020b].

Finally, I further improve the multi-channel neural net classifier by applying it to label tweets and using the newly labeled tweets to augment the training set and retrain the classifier. The whole process goes for several iterations. The overview of the full system is shown in Figure 1.7. The evaluation on two hurricane datasets, hurricane *Harvey* and *Florence*, shows that the rapidly trained weakly supervised systems using the novel data labeling method outperforms the supervised learning approach requiring thousands of carefully annotated tweets created in over 50 person-hours.

1.3 Thesis Outline

The outline of this dissertation is summarized as follows.

- Chapter 2 provides relevant background and previous work. I will discuss different event representations in previous work and the event representation in this dissertation, a summary of weakly-supervised approaches, classical event-event relation identification tasks, existing methods of acquiring event knowledge, relevant work of narrative identification, and other related NLP research.

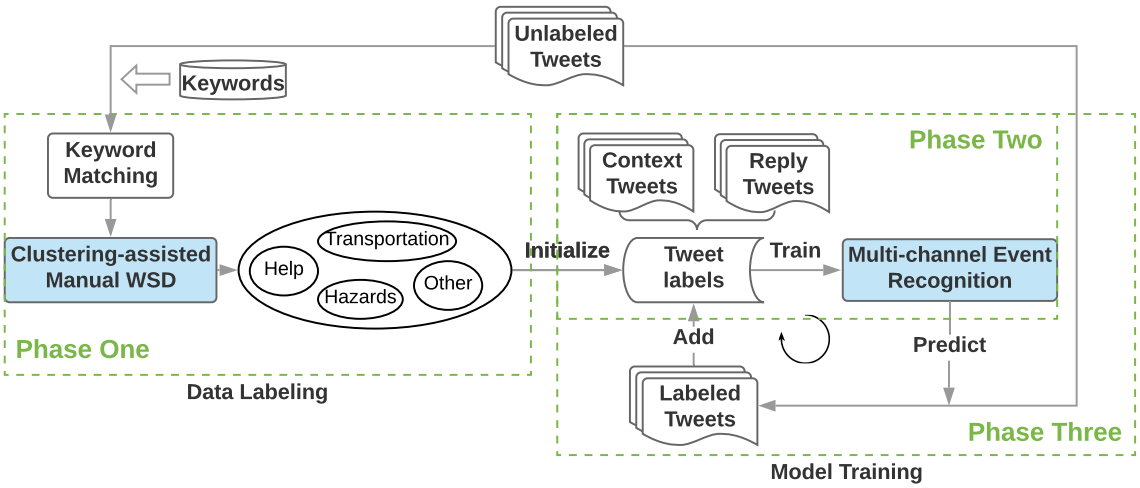

Figure 1.7: Overview of the Weakly-supervised Event Recognition System for Disaster Management. Reprinted with permission from Yao et al. [2020b].

- Chapter 3 introduces my first weakly-supervised approach to train a temporal relation classifier and acquire event temporal knowledge simultaneously. In this chapter, I will first discuss how to prepare event temporal pair seeds and candidates. Next, I consider two variations of sentential contexts, i.e., local windows or dependency paths in contextual classifier training. Evaluation shows my full system (argument generalization together with dependency path contexts) performs the best on acquiring event temporal pairs.

- Chapter 4 presents my second weakly-supervised approach to acquire event temporal knowledge by identifying narratives. I will discuss some key elements in narratology that can be used to identify narratives. My learning method contains two stages. Stage one uses bootstrapping learning to extract narrative paragraphs based on key elements of narratives. Stage two applies Causal Potential metric to distill event temporal knowledge from narratives. Finally, I evaluate acquired narrative paragraphs and event temporal knowledge separately.

- Chapter 5 demonstrates a weakly-supervised approach to acquire subevent knowledge. To collect high-quality weak supervision to train a contextual classifier, I conduct definition-guided semantic check to remove spurious subevent pairs that only include two temporally

overlapping. Next, I will talk about how to train a contextual classifier that can recognize subevent relations by fine-tuning a BERT model. Finally, the section of intrinsic and extrinsic evaluations show acquired subevent knowledge are of high quality and can be used to improve three event relation classification tasks and the discourse parsing task.

- In Chapter 6, I present details of how to apply a similar approach of Chapter 5 to causal knowledge acquisition. I next discuss how to incorporate different event knowledge into an event knowledge graph and how to learn event distributed representations using knowledge graph embedding. This chapter includes a visualization of learned event embeddings to give insights on how learned event embeddings are semantically different from traditional word embeddings.

- Chapter 7 proposes a fine-grained event recognition system for detecting life-threatening emergencies in hurricanes on Social Media. This Chapter illustrates three phrases one by one. In phrase one, I apply the novel clustering-assisted manual Word Sense Disambiguation (WSD) to label data. Phase two conducts multi-channel classification based on context posts from the same authors and replies from other authors. Phase three further improves the coverage of the multi-channel classifier through bootstrapping learning. Finally, evaluation experiments show that my approach can significantly outperform multiple baseline systems.

- Chapter 8 summarizes the conclusions that we can draw from the dissertation. Following the conclusions, I will discuss promising future topics that may move forward weakly-supervised learning on event knowledge acquisition and event detection.

2. BACKGROUND AND RELATED WORK

This dissertation focuses on applying weakly-supervised learning approaches to acquire event relational knowledge and detecting fine-grained events on social media during natural disasters. In the following sections, We first introduce what are events and the event representation in my event knowledge learning. We then review weakly-supervised learning approaches that are commonly used in NLP research. Next, we discuss conventional event relation identification tasks that mainly contain temporal relation identification, subevent relation identification, and causal relation identification between events. Moreover, we talk about existing knowledge bases that contain event knowledge as well as corresponding approaches to construct those knowledge bases. Regarding the second part of this dissertation - fine-grained event detection on social media, we review previous work related to social sensing on social media platforms, with emphasis on event sensing. My research also benefits from recent advances of neural representation learning in NLP, so we lastly cover recent work on contextual word embedding learning, knowledge graph embedding learning in the last section.

2.1 Background: Event and Event Representations

"An event is something that happens. People use events to describe all the things that are happening in a particular situation." However, there is a range of different representations of event [Bies et al., 2016]. Most commonly used event representation is based on Automatic Content Extract (ACE) [Doddington et al., 2004] where each event mention annotation contains an event trigger, the event type/subtype, and participating event arguments together with time expressions. Other previous event representations are "Light Entities, Relations, and Events (Rich ERE)", "Rich Entities, Relations, and Events (Light ERE)" [Song et al., 2015], "Event Nugget (EN)" [Mitamura et al., 2015], "Richer Event Descriptions (RED)" [O'Gorman et al., 2016], etc. Light ERE reduces entity and relation types with fewer attributes, while Rich ERE expands event ontology, adds realis attribute (Actual, Generic, Other) and other event argument types. "Event Nugget" simplifies event

representation as a tuple of an event trigger, event type/subtype, and realis attribute. RED focuses more on annotating event-event relations (e.g., coreference, temporal and causal relations) in a document, so RED annotates all occurrences and timeline-relevant states as events but without any arguments. We can see that previous work mainly represents an event as a trigger together with multiple participants (arguments). In this dissertation, I represent an event as a trigger word (normally a noun or a verb word) together with some syntactic elements (e.g., subject and object) to support weakly-supervised learning.

2.1.1 Event Representation in this Dissertation

My weakly-supervised learning approach relies on identifying event pairs that tend to unambiguously show a particular event relation. However, an event word can refer to a general type of events or more than one type of events, and therefore has varied meanings depending on contexts. To make individual events expressive and self-contained, I find and attach arguments to each event word to form event phrases. Specifically, I consider both verb event phrases (Section 2.1.1.1) and noun event phrases (Section 2.1.1.2). I 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, I do not consider event pairs in which neither event has an argument.

2.1.1.1 Verb Event Phrases

To ensure a good coverage of regular event pairs, I consider all verbs[1] as event words except reporting verbs[2]. The thematic patient of a verb refers to the object being acted upon and is essentially part of an event, therefore, I first include the patient of a verb in forming an event phrase. I use Stanford dependency relations [Manning et al., 2014a] to identify the direct object of an active verb or the subject of a passive verb. The agent is also useful to specify a event especially for a intransitive verb event, which does not have a patient. Therefore, I include the agent of a verb event in an event phrase if its patient was not found. Agents are usually the syntactic subject of an active

[1] I used POS tags to detect verb events.

[2] Reporting verbs, such as "said", "told" and "added", are commonly seen in news articles. I determined that most of event pairs containing a reporting verb are not very interesting and informative and I therefore discarded these event pairs.

verb or *by* prepositional object of a passive verb.

For instance, in the sentence *"They win the lottery."*, the verb *win* can refer to various *win* events, but with its direct object, *win lottery* refers to a specific type of event. For another instance, *"Water evaporates when it's hot."*, the verb *evaporates* itself is not very meaningful without contexts, but after including its subject, the event *water evaporates* becomes self-contained. If neither a patient nor an agent was found, I include a prepositional direct object of a verb in the event representation to form an event phrase.

2.1.1.2 Noun Event Phrases

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

2.1.1.3 Generalizing Event Arguments Using Named Entity Types

Including arguments into event representations generates specific event phrases though. In order to obtain generalized event phrase forms, I replace specific name arguments with their named entity types [Manning et al., 2014a]. I also consider replacing pronouns with their types, but concerned with poor quality of full coreference resolution, I only replace personal pronouns with their type PERSON. I observed that this strategy greatly improves generality of event phrases and facilitates the bootstrapping learning process.

2.2 Weakly-supervised Learning in NLP

Traditional supervised learning approaches strongly rely on labeled data that are limited in size and diversity to train the model. Thus, weakly-supervised learning and semi-supervised learning are developed to address this problem. Weakly-supervised learning is closely related to semi-supervised learning, but semi-supervised learning addresses this problem by using abundant unlabeled data, together with labeled data, to build better models.

Two most popular algorithms for semi-supervised learning are self-training and co-training. Riloff et al. [2003] introduces two bootstrapping algorithms, Meta-bootstrapping and Basilisk. Two algorithms exploit mining patterns to learn words in a specific semantic category. Meta-bootstrapping and Basilisk start with syntactic templates as extraction patterns and then calculates a score for each pattern based on the seed words and save the best patterns to a pool. Next, a pattern in the pool can extract candidate words and score those words based on their collective association with the seed words. Top words are labeled automatically as the semantic category targeted. Co-training [Blum and Mitchell, 1998, Zhou and Li, 2005] assumes that each instance can be viewed using two dissimilar feature sets that provide different, complementary information about the instance. Preferably, two views are conditionally independent and each view is sufficient to predict the class of each instance. Co-training starts with learning separate classifiers over each feature set. Predictions on unlabeled data with highest confidence scores are added to augment labeled data. Then, it iteratively rebuilds each classifier using an augmented labeled set.

In contrast, my proposed weakly-supervised approaches adopt and develop bootstrapping learning idea to event relational knowledge acquisition and fine-grained event detection on social media. My approaches rely on **weak supervision** that is generated by applying linguistic patterns, recognizing narrative key elements, and clustering-assisted Word Sense Disambiguation, which does not require any labeled instances.

2.3 Event-Event Relation Identification

Since my event knowledge acquisition focuses on event relational knowledge, I will review some previous work on event-event relation identification in this section.

2.3.1 Temporal Relation Recognition

Supervised temporal relation classifier has been well studied in previous work. Most of existing temporal relation classifiers were learned in a supervised manner and depend on human annotated data. In the TempEval campaigns [Verhagen et al., 2007, 2010, UzZaman et al., 2013], various classification models and linguistic features [Bethard, 2013, Chambers et al., 2014, Llorens et al., 2010, D'Souza and Ng, 2013, Mirza and Tonelli, 2014b] have been applied to identify temporal relations between two events. For example, a recent study by [D'Souza and Ng, 2013] applied sophisticated linguistic, semantic and discourse features to classify temporal relations between events. They also included 437 hand-coded rules in building a hybrid classification model. Similarly, Mirza and Tonelli [2014b] shows basic information on the position, the attributes of events, as well as other information obtained from external lexical resources outperforms sophisticated features. CAEVO, a CAscading EVent Ordering architecture by Chambers et al. [2014], applied a sieve-based architecture for event temporal ordering. CAEVO is essentially a hybrid model as well. While the first few sieves are rule based and deterministic, the latter ones are machine learned using human annotated data. More recently, Choubey and Huang [2017] uses three bi-directional LSTMs to get the embeddings of three sequences (i.e., word forms, POS tags, and dependency relations) of context words that align with the dependency path between two event mentions. The final fully connected layer maps the concatenated embeddings of all sequences to 14 fine-grained temporal relations.

2.3.2 Subevent Relation Recognition

Only a few studies have focused on identifying subevent relations in text. Most of existing work on recognizing subevent relations is learned in a supervised manner that depends on human annotated data. [Araki et al., 2014] built a logistic regression model to classify the relation be-

tween 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 on two datasets using a linear SVM classifier, by introducing several new features, in particular discourse features (i.e., rhetorical structure, reported speech, etc.) and narrative features (i.e., non-major mentions).

In addition, I want to make aware of previous research that study subevents for event tracking applications specifically for social media (e.g., Twitter) applications Shen et al. [2013], Meladianos et al. [2015], Pohl et al. [2012], in terms of both its definition of subevents and methodologies. For example, in previous research by Shen et al. [2013], a subevent is defined as a topic that is discussed intensively in the Twitter stream for a short period of time before fading away. Accordingly, the subevent detection method relies on modeling the "burstiness" and "cohesiveness" properties of tweets in the stream.

2.3.3 Causal Relation Recognition

Detecting causality between events is challenging and has been addressed by several pilot studies [Girju, 2003, Bethard and Martin, 2008, Riaz and Girju, 2010, Do et al., 2011, Riaz and Girju, 2013] via supervised classification models. Recently, researchers have also employed unsupervised causality detection. Riaz and Girju [2010] proposed Effect-Control Dependency (ECD) metric to determine causality. Later, Do et al. [2011] used PMI and ECD to predict causal relation in verbal and nominal event pairs. Riaz and Girju [2013] proposed a set of metrics (i.e., Explicit Causal Association (ECA), Implicit Causal Association (ICA) and Boosted Causal Association(BCA)) to learn the likelihood of causal relations between verbs and identified three categories of verb pairs: strongly causal, ambiguous and strongly non-casual. Recently, Mirza and Tonelli [2014a] presented annotation guidelines and annotated explicit causality between events in Timebank. With

the resulted corpus, called Causal-TimeBank, they built supervised models to identify causal relations. Then Mirza and Tonelli [2016] proposed a sieve-based method called CATENA, to perform joint temporal and causal relation extraction, exploiting interactions between temporal and causal relations. Ning et al. [2018] trained a constrained conditional model (CCM) by combining temporal and causal relations. Their constrains include 1) a cause must happen before the effect; 2) symmetry constrains, e.g., *work* after *graduation* implies *graduation* before *work*; 3) transitivity constraints, e.g., *order* before *eat* together with *eat* before *pay* can imply *order* before *pay*.

2.4 Event Knowledge Acquisition

Most existing knowedge bases such as Freebase [Bollacker et al., 2008] and Probase [Wu et al., 2012] often contain rich knowledge about entities (e.g., people, country, movie, etc.). In this section, I will talk about a few knowledge bases that contain some event knowledge and the corresponding methods to acquire them.

2.4.1 Crowdsourcing Methods

FrameNet project [Baker et al., 1998] that is launched in 1997 aims to build both human and machine readable knowledge resources. A frame is a event (or state) representation involving various participants (frame elements), relations between frames (frame relations) and other conceptual roles. More than 200,000 sentences are manually annotated and linked to more than 1,200 semantic frames. Similarly, inspired by the distributed projects on the Web, ConceptNet [Havasi et al., 2007] that is built upon Open Mind Common Sense (OMCS) website asks human volunteers to write down common sense knowledge of daily activities. Among its predefined interlingual relations, three are related to events, i.e., *HasSubevent, HasLastSubevent, HasFirstSubevent*. To support commonsense inference on events, Event2Mind [Rashkin et al., 2018] crowdsources 25K event phrases covering everyday events and situations from stories, blogs, and Wiktionary idioms and proposes a new task that aims to generate descriptions of intents and reactions of event participants.

2.4.2 Pattern-based methods

Related to my approach, Pattern based methods have been applied to acquire event pairs in a specific semantic relation. Specifically, VerbOcean [Chklovski and Pantel, 2004a] extracted fine-gained semantic relations between verbs including the happens-before relation using lexico-syntactic patterns. It turns out that the temporal relation patterns used in VerbOcean (e.g., "to X and then Y") are too specific and not capable of identifying many event pairs that are rarely seen in one of the specified patterns. The first approach I proposed (Chapter 3) can gradually learn a collection of regular event pairs using more diverse contexts by using the weakly supervised trained temporal relation classifier to recognize diverse contexts that describe a particular temporal relation. In the meanwhile, these prior works are limited to identifying temporal relations within individual sentences. In contrast, my approach to acquire event temporal knowledge by identifying narratives (Chapter 4) was inspired by the double temporality property of the narrative genre and is designed to acquire temporal relations across sentences in a narrative paragraph. Evaluation shows that my approach induces very different event pairs from VerbOcean. Based on my comparison, only 1% out of all event pairs acquired by my two approaches can be found in VerbOcean [Chklovski and Pantel, 2004a].

The pilot research on subevent knowledge acquisition Badgett and Huang [2016] relies on two heuristics, 1. subevent phrases often occur in conjunction constructions as a sequence of subevent phrases, and 2. they often appear in sentences that start or end with characteristic phrases such as "media reports" and "witness said". and only learns several hundred subevent phrases for one type of parent events, civil unrest events.

2.4.3 Generation-based Methods

More recently, [Bosselut et al., 2019, Sap et al., 2019] investigated a semi-automatic approach to generate commonsense knowledge on events. Specifically, they train generative language models (based on LSTM or Transformer) to generate subevent knowledge among many other types of commonsense knowledge.

2.4.4 Hybrid Methods

Knowlywood [Tandon et al., 2015] presents a pipeline with semantic parsing and knowledge distillation from movie scripts and narrative texts to acquire semantic frames about human activities. Knowlywood first uses semantic parsing to extract human activities (verb+object) and apply Probabilistic Soft Logic (PSL) to derive connections between activity frames, such as hypernyms, similarities, and temporal order. ASER [Zhang et al., 2020] builds a large-scale eventuality knowledge graph of 194M eventualities and 64M relation edges among them. ASER knowledge extraction has two big steps. It first preprocesses raw texts with the dependency parser and applies predefined patterns to extract all eventualities. Next, by applying a discourse parser trained on PDTB [Prasad et al., 2008] dataset, ASER extracts relations between eventualities based on less ambiguous connectives. Compared to my weakly-supervised approaches, ASER strongly relies on the performance of discourse parser pretrained on PDTB, which may limit its coverage and accuracy of acquired eventuality knowledge.

2.5 Narratives Identification

Our design of the overall event knowledge acquisition also benefits from recent progress on narrative identification. Gordon and Swanson [2009] annotated a small set of paragraphs presenting stories in the ICWSM Spinn3r Blog corpus [Burton et al., 2009] and trained a classifier using bag-of-words features to identify more stories. [Ceran et al., 2012] trained a narrative classifier using semantic triplet features on the CSC Islamic Extremist corpus. My weakly supervised narrative identification method is closely related to Eisenberg and Finlayson [2017], which also explored the two key elements of narratives, the plot and the characters, in designing features with the goal of obtaining a generalizable story detector. But different from this work, my narrative identification method does not require any human annotations and can quickly adapt to new text sources.

2.6 Learning Script Knowledge

Temporal event knowledge acquisition is related to script learning [Chambers and Jurafsky, 2008], where a script consists of a sequence of events that are often temporally ordered and repre-

sent a typical scenario. However, most of the existing approaches on script learning [Chambers and Jurafsky, 2009, Pichotta and Mooney, 2016, Granroth-Wilding and Clark, 2016] were designed to identify clusters of closely related events, not to learn the temporal order between events though. For example, Chambers and Jurafsky [2008, 2009] learned event scripts by first identifying closely related events that share an argument and then recognizing their partial temporal orders by a separate temporal relation classifier trained on the small labeled dataset TimeBank [Pustejovsky et al., 2003]. Using the same method to get training data, Jans et al. [2012], Granroth-Wilding and Clark [2016], Pichotta and Mooney [2016], Wang et al. [2017b] applied neural networks to learn event embeddings and predict the following event in a context. Distinguished from the previous script learning works, I focus on acquiring event pairs or longer script-like event sequences with events arranged in a complete temporal order. Crowdsourcing has been used for acquiring script-like event chains or stories [Regneri et al., 2010, Frermann et al., 2014, Modi and Titov, 2014, Modi et al., 2016, Mostafazadeh et al., 2016]. Regneri et al. [2010] collected script-like event chains by asking Amazon Mechanical Turk (AMT) to write down typical event sequences in a given scenario (e.g., shopping or cooking). This collection of *Event Sequence Descriptions* (ESDs) is then used to construct a *temporal script graph* using the Sequence Alignment algorithm. Similarly, InScript [Modi et al., 2016] predefined 10 scenarios and collected 1,000 stories that Instantiate Script structure in natural language. Mostafazadeh et al. [2016] introduced a story-cloze task based on a collection of five-sentence stories. Give four sentences as context, the task requires choosing the correct ending sentence. Interestingly, the evaluation shows that my approach can yield temporal event knowledge that covers 48% of human-provided script knowledge.

2.7 Event Detection on Social Media

Previous research for Twitter event detection mostly focuses on unsupervised statistical approaches, which can be categorized into three main streams. 1) Identifying burst topics. For example, inspired by congestion control algorithms, TwitInfo Marcus et al. [2011] used a weighted moving average and variance model to detect peaks of terms in Twitter data to track an event. Alvanaki et al. [2011] examined the correlation between hashtag pairs within a time window and

detected sudden increases of correlation as an indicator of an emergent topic. 2) Probabilistic topic modeling. For example, Latent Event and Category Model (LECM) Zhou et al. [2015], Cai et al. [2015] modeled each tweet as a joint distribution over a range of features (e.g., text, image, named entities, time, location, etc.). 3) Unsupervised clustering approaches. New tweets are determined to merge into an existing cluster or form a new cluster based on a threshold of similarity Becker et al. [2011], and events are summarized from clusters using metrics such as popularity, freshness and confidence scores.

Supervised classification approaches are used for Twitter event detection Zhang et al. [2019]. Different classification methods, Naive Bayes Sankaranarayanan et al. [2009], Support Vector Machines Sakaki et al. [2010], Decision Trees Popescu et al. [2011], and Neural Networks Caragea et al. [2016], Nguyen et al. [2017] have been used to train event recognizers using human annotated Twitter messages. However, annotating a large number of Twitter messages for a new disaster is time-consuming, and systems trained using old labeled data may be biased to only detect information specific to one historical disaster (e.g., local road names, local authorities, etc.). In contrast, the weakly supervised classification approach I propose does not require slow brewed training instances annotated one by one, and can quickly label data and train event recognizers from scratch for a newly happened disaster.

2.8 Other Related NLP Research

2.8.1 Recent Advances of Contextual Word Representation

Conventional word embeddings such as GloVe [Pennington et al., 2014], word2vec [Mikolov et al., 2013a], and fastText [Mikolov et al., 2018] map each word in a vocabulary into a continuous space so that similar words (e.g., synonyms) are close in that space. These embeddings are pretrained on large text corpora based on word co-occurrence. However, these embeddings are trained globally which means one word corresponds to one embedding vector regardless of different contexts the word may occur in. Recently, ELMo embedding model [Peters et al., 2018] generalizes conventional word embeddings by pretraining a language model to extract context-

dependent representations in a sentence. ELMo pretrains a forward LSTM language model and a backward LSTM language model, which maximizes the conditional probability of $token_i$ given a sequence of previous tokens $token_1, token_2, ..., token_{i-1}$. After training, the hidden representations at $token_i$ of forward LSTM and backward LSTM are concatenated as the contextual embedding of $token_i$. BERT [Devlin et al., 2019] further develops this idea and designs two objectives to pretrain a Transformer [Vaswani et al., 2017] encoder. The first objective is masked language modeling, where randomly selected tokens of a given input sentence are masked, and BERT tries to maximize the conditional probability of masked tokens given the corrupted sentence. The second objective is next-sentence-prediction, where BERT predicts which candidate sentence is the next sentence of the given sentence.

Previous work [Peters et al., 2018, Devlin et al., 2019, Yang et al., 2019, Raffel et al., 2019] demonstrates that contextual embeddings pretrained on large language corpora achieve great performance improvement on various NLP tasks, such as text classification, machine translation, and text generation. Clark et al. [2019] has shown contextual embeddings can capture syntactic dependency and semantic properties of words more effectively in different contexts.

2.8.2 Knowledge Graph Embedding Approaches

After acquiring different event relational knowledge, this dissertation also investigates embedding models to generate a distributed representation of each event phrase. Intuitively, knowledge graph embedding models try to map each graph node and relation type into a continuous vector space so that the inherent structure of the knowledge graph is preserved. Wang et al. [2017a] categorizes knowledge graph embedding models into *translated distance models* and *semantic matching models*. Suppose we have a knowledge graph stored as a collection of tuples (head, relation, tail), TransE [Bordes et al., 2013] map each node and relation type into a low-dimentional space $(\mathbf{h}, \mathbf{r}, \mathbf{t})$ so that $\mathbf{p} + \mathbf{r} \approx \mathbf{c}$. The score function is defined as the distance function $f_r(h, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{1/2}$. However, TransE has flaws when deal with N-to-1, 1-to-N, and N-to-N relations, therefore, TransH [Wang et al., 2014] introduce an idea that allows an entity to have different representations when attending different relations. TransR [Lin et al., 2015] use a similar

idea with TransH but applies relation-specific spaces instead of hyperplanes.

3. A WEAKLY SUPERVISED APPROACH TO TRAIN TEMPORAL RELATION CLASSIFIERS AND ACQUIRE REGULAR EVENT PAIRS SIMULTANEOUSLY[1]

The first weakly supervised approach focuses on detecting "before/after" temporal relations and apply patterns to first extract thousands of regular event pairs and train a contextual temporal relation classifier simultaneously. Figure 3.1 illustrates how the bootstrapping system works. I first populate seed regular event pairs in the text corpus and identify sentences that contain a regular event pair as training instances. I train a contextual temporal relation classifier, using Convolutional Neural Nets (CNNs), to identify specific contexts describing a temporal *after (before)* relation. I then apply the classifier to the corpus to identify new sentences that describe a particular temporal relation, from which new regular event pairs can be extracted. Note that the classifier is only applied to sentences that contain a candidate regular event pair. The bootstrapping process repeats until the number of newly identified regular event pairs is less than 100.

3.1 Regular Event Pair Candidates

Considering that it is not feasible to test all possible pairs of events in Gigaword and often two events that co-occur in a sentence have no temporal relation. In order to narrow down the search space, I identify candidate event pairs which are likely to have temporal relations.

Two strategies are used to identify candidate event pairs. First, by intuition, if two event phrases co-occur (within a sentence) many times, the likelihood of the two events being related and having a temporal relation should be higher compared to event phrases that rarely co-occur. Therefore, I select event phrase pairs that co-occur within a sentence for more than 100 times as candidate event pairs. Second, I use two specific temporal relation patterns, EV_A *after (before)* EV_B, that explicitly indicate two events are in a after (before) relation. I extract an event pair as a candidate regular pair if it occurs three or more times with one of the patterns in the text corpus. The

[1]Reprinted with permission from "A Weakly Supervised Approach to Train Temporal Relation Classifiers and Acquire Regular Event Pairs Simultaneously" by Wenlin Yao and Saipravallika Nettyam and Ruihong Huang, 2017. Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP), pp.803-812.

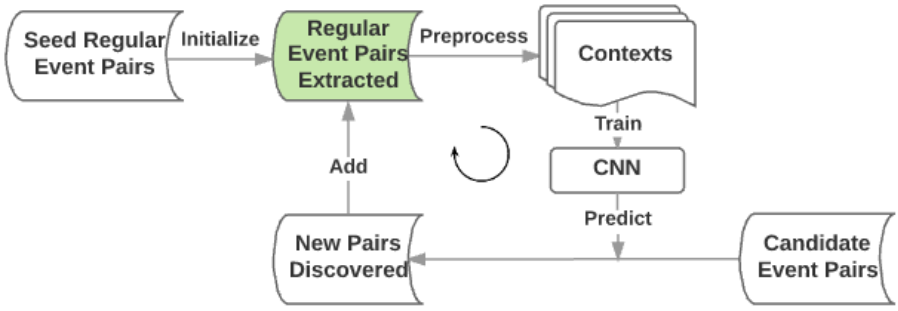

Figure 3.1: Overview of the Bootstrapping System for Temporal Knowledge Acquisition. Reprinted with permission from Yao et al. [2017].

assumption is that if a pair of events shows a particular temporal relation regularly, it is likely to be seen in the above textual patterns as well. Specifically, I extract the governor and dependent word of the dependency relation *prep_after (prep_before)* in the annotated English Gigaword [Napoles et al., 2012] and check whether each word is an event[2]. If yes, I form an event phrase for each event and obtain an event pair. In addition, I expect regular event pairs to occur mostly in a single temporal order, either *before* or *after*, and discard event pairs that have showed mixed temporal orders. Specifically, a regular event pair is required to occur in a particular temporal relation more than 90% of times.

Overall by applying the two strategies, I obtained a candidate event pair pool that consists of 40,278 event pairs.

3.2 Bootstrapping both Regular Event Pairs and a Temporal Relation Classifier

While I used the whole Gigaword [Napoles et al., 2012] to identify regular event pairs, I only use the New York Times section of Gigaword for bootstrapping learning.

3.2.1 Regular Event Pair Seeds

In order to ensure high quality of seed pairs, I only consider event pairs that have occurred in explicit temporal relation patterns, EV_A *after (before)* EV_B, as seed event pairs. Furthermore, I require each seed regular event pair to have occurred in a temporal relation pattern for at least ten

[2]Note I consider any verb and a noun that is in my noun event list as an event.

times. Specifically, I identified 2,110 seed regular event pairs using the Gigaword corpus[3].

3.2.2 Contextual Temporal Relation Classification

I use a neural net classifier to capture compositional meanings of sentential contexts and avoid tedious feature engineering. Specifically, I used a Convolutional Neural Net (CNN) as my classifier, inspired by recent successes of CNN models in various NLP tasks and applications, such as sentiment analysis [Kalchbrenner et al., 2014, Kim, 2014], sequence labeling [Collobert et al., 2011] and semantic parsing [Yih et al., 2014]. As shown in figure 3.2, my CNN architecture is a slight variation of the previous models as described in [Kim, 2014, Collobert et al., 2011]. It has one convolutional layer with 100 hidden nodes, one pooling layer and one fully connected softmax layer.

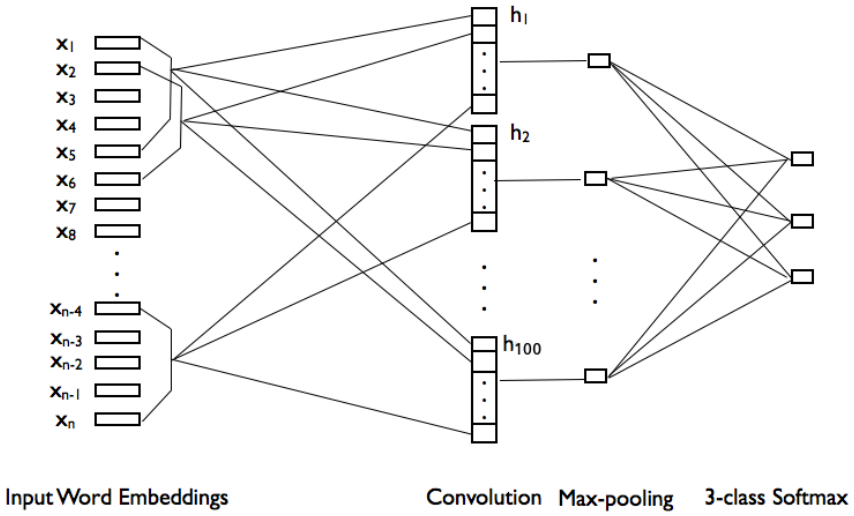

Figure 3.2: CNN Model Architecture. Reprinted with permission from Yao et al. [2017].

The input are word embeddings of an array of sentential context words.

A convolution filter is applied to a sliding window of every h words to provide input for each hidden node. I use Rectified Linear Unit (ReLU) as the non-linear activation function.

[3]By populating seed regular event pairs in the New York Times section of the Gigaword corpus, I extracted 7191 sentences and 11339 sentences that contain an event pair in a "before" and "after" temporal relation respectively.

I next apply a max-pooling operation to take the maximum value over a feature map. The final softmax layer output probability distributions over three classes (AFTER, BEFORE and OTHER) indicating the temporal relation between a pair of events in a sentence. Specifically, the temporal relations are defined with respect to the textual order the two events are presented in a sentence. If the first event is temporally BEFORE the second event as described in a sentence, this instance will be labeled as BEFORE. Otherwise if the first event is temporally AFTER the second event as described in a sentence, the instance will be labeled as AFTER. The class OTHER is to capture all the rest contexts that may describe a temporal relation other than *after (before)* or do not describe a temporal relation.

In my experiments, I use pre-trained 300-dimention word2vec word embeddings [Mikolov et al., 2013b] that are trained on 100 billion words of Google News and I use a filter window size of 5. In training, I used stochastic gradient descent with Adadelta update rule (Zeiler, 2012) and mini-batch size of 100, in addition, I applied dropout [Hinton et al., 2012] with rate $p = 0.5$ to avoid overfitting of the CNN model. I also randomly selected 10% of the training data as the validation set and chose the classifier with the highest validation performance within the first 10 epochs.

3.2.2.1 Sentential Contexts: Local Windows v.s. Dependency Paths

I explore two types of contexts, local windows v.s. dependency paths, in order to identify contexts that effectively describe temporal relations between two events.

First, the local window based context for an event pair includes five words before the first event, five words after the second event and all the words between the two events. Note that two event phrases can be arbitrarily far from each other and long contexts are extremely challenging for a classifier to capture. In my experiments, I only consider sentences where two event mentions are at most 10 words away.

Second, I observed that not every word between two events is useful to predict their temporal relation. In order to concentrate on relevant context words,

I further construct dependency path[4] based context representation. Specifically, considering a dependency tree as an undirected graph, I use breadth-first-search to extract a sequence of words connecting the first event word to the second event word. In addition, to capture important information in certain syntactic structures such as conjunctions, I extract children nodes for each word in the path. Finally, I sort extracted words according to their textual order in the original sentence and the sorted sequence of words is provided as an input to the CNN classifier.

3.2.2.2 Negative Training Instances

Reasonably, most sentences in a corpus do not contain an event pair that is in a temporal "before/after" relation. Therefore, I use negative instances that are 10 times of the total number of positive training instances (i.e., sentences that contain an event pair in a *after (before)* relation). Specifically, I require a negative instance to contain an event pair that does not appear in seed pairs nor the candidate event pair set. I randomly sampled negative instances satisfying the condition. Then these deemed negative instances were labeled as the class OTHER, a class that compete with the two temporal relation classes, BEFORE and AFTER.

Systems	0 (Seeds)	1	2	3	4	5	Total
Basic System	1057	213	102	48	–	–	1420
+ Arg Generalization	2110	638	323	81	–	–	3152
+ Dependency Path Contexts (Full System)	2110	1230	555	288	156	62	4401

Table 3.1: Number of New Regular Event Pairs Generated after Each Bootstrapping Iteration. Reprinted with permission from Yao et al. [2017].

3.2.3 New Regular Event Pair Selection Criteria

Recall that regular event pairs are event pairs that tend to show a particular temporal relation despite of their various contexts. Therefore, I identify a candidate event pair as a new regular event pair if majority of its sentential contexts, specifically 60% of contexts, were consistently labeled

[4]Stanford CoreNLP [Manning et al., 2014a] were used to generate dependency trees.

as a particular temporal relation (*after* or *before*) by the CNN classifier. In addition, I require that at least 15 instances of a regular event pair have been labeled as the majority temporal relation. In order to control semantic drift [McIntosh and Curran, 2009] in bootstrapping learning, I increase the threshold by 5 after each iteration.

Furthermore, in order to filter out ambiguous event pairs that can be in either *before* or *after* temporal order depending on concrete contexts, I require the absolute difference between number of instances labeled as AFTER and labeled as BEFORE to be greater than a ratio of the total number of instances, specifically, I set the ratio to be 40%.

3.3 Evaluation

My bootstrapping system learned regular event pairs as well as a contextual temporal relation classifier. I evaluate each of the two learning outcomes separately.

3.3.1 Regular Event Pair Acquisition

3.3.1.1 System Variations

I compare three variations of my system:

Basic System: in the basic system, I did not apply event argument generalization as described in section 2.1.1.3. In addition, I use local window based sentential contexts as input for the classifier.

+ *Arg Generalization*: on top of the basic system, I apply event argument generalization.

+ *Dependency Path Contexts (Full System)*: in the full system, I apply event argument generalization and use dependency path based sentential contexts as input for the classifier.

Table 3.1 shows the number of regular new pairs that were generated after each bootstrapping iteration by each of the three systems. First, we can see that event argument generalization is useful in obtaining roughly two times of seed regular event pairs. Second, event argument generalization is useful in recognizing additional regular event pairs in bootstrapping learning as well. Third, dependency path based sentential contexts are effective in capturing relevant sentential contexts for temporal relation classification, which enables the bootstrapping system to maintain a learning momentum and learn more regular event pairs.

Systems	Seed Pairs	New Pairs
Basic System	0.73	0.55
+ Arg Generalization	0.71	0.63
+ Dependency Path Contexts		0.67

Table 3.2: Accuracy of 100 Randomly Selected Event Pairs. Reprinted with permission from Yao et al. [2017].

3.3.1.2 Accuracy of Regular Event Pairs

For each of the three system variations, I randomly selected 50 pairs from seed regular event pairs and 50 from bootstrapped event pairs[5] and asked two human annotators to judge the correctness of these acquired regular event pairs.

Specifically, for each selected event pair, I ask two annotators to label whether a temporal AFTER or BEFORE relation exists between the two events. In addition to the two temporal relation labels, I provide the third category OTHER as well. I instruct annotators to assign the label OTHER to an event pair if the two events (i) generally have no temporal relation, (ii) have a temporal relation other than AFTER or BEFORE, or (iii) one or both mentions do not refer to an event at all.[6] For each event pair, only one label is allowed. Before the official annotation, I trained the two annotators with system generated event pairs for several iterations. The event pairs I used in training annotators are different from the final event pairs I used for evaluation purposes.

Table 3.2 shows the accuracy of regular event pairs learned by each system variation. I determine that an event pair is correctly predicted by a system if the system predicted temporal relation is the same as the label that has been assigned by both of the two annotators. The overall kappa inter-agreement between the two annotators is 72%. We can see that with event argument generalization, the quality of acquired seed regular event pairs is roughly equal to that using specific name arguments. Furthermore, because I obtained two times of seed event pairs after using event argument generalization, the second and third bootstrapping systems received more guidance and

[5]The seed pairs for the second and the third system are the same, so I evaluate the same 50 randomly selected seed pairs for the two systems.

[6]This can happen due to Part-Of-Speech errors or ambiguous event words.

Common Sense	PERSON **worked** ← **graduation** **career** → **announced** retirement **wash** hands → **eating** PERSON **returned** ← **visit**
Politics	government be **formed** ← **elections** **fled** mainland ← **losing** war **imposed** sanctions ← **invasion** of LOCATION LOCATION **split** ← **war**
Business	**reached** agreement ← **negotiations** **hosted** banquet ← **meeting** **trading** → stock **closed**
Health	**cause** of death ← **cancer** PERSON be **hospitalized** ← **suffering** stroke PERSON **died** ← **admitted** to hospital
Sports	**games** → **ended** season PERSON be **sidelined** ← **undergoing** surgery PERSON be **suspended** ← **testing** for cocaine PERSON **returned** ← **recovering** from injury
Crime	**shooting** → PERSON be **arrested** **spending** in jail → PERSON be **released** PERSON be **arrested** ← **bombings** driver **fled** ← **accident**

Table 3.3: Examples of Learned Regular Event pairs. Reprinted with permission from Yao et al. [2017]. → represents *before* relation and ← represents *after* relation.

continued to learn regular event pairs with a high quality. In addition, using dependency path based sentential contexts enables the classifier to further improve the accuracy of bootstrapped regular event pairs.

3.3.1.3 Examples and Constructed Knowledge Graphs

I have learned around 4,400 regular event pairs that are rich in commonsense knowledge and domain specific knowledge for domains including politics, business, health, sports and crime. Table 3.3 shows several examples in each category.

In addition, related event pairs form knowledge graphs, figure 3.3 shows two examples. The first one describes various scenarios that cause deaths while the second one describes contingent relations among events specific in sports.

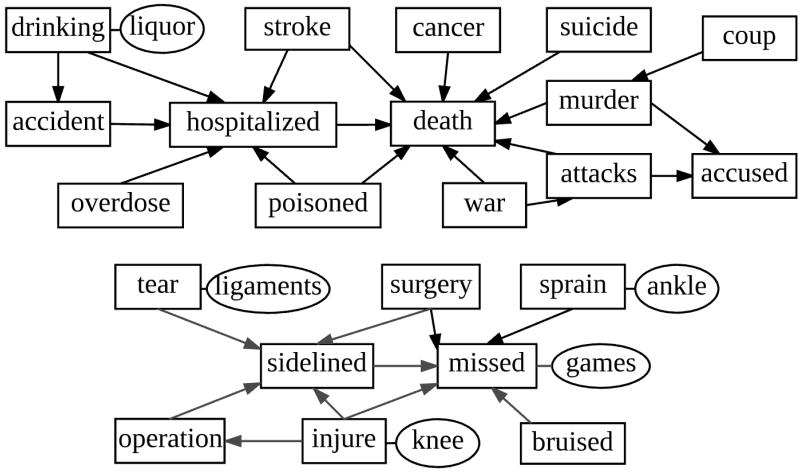

Figure 3.3: Example Temporal Event Knowledge Graphs. → denotes the *happens_before* temporal relation. Reprinted with permission from Yao et al. [2017].

3.3.1.4 Causally Related Events

I observed that a large portion of the learned regular event pairs are both temporally and causally related. I adopt the force dynamics theory and determine that two events are causally related if one event causes, enables or prevents the other event to happen. Then I asked two annotators [7] to annotate causal relations for the same set of 100 randomly selected regular event pairs generated by the full bootstrapping system. Surprisingly, out of 69 event pairs that have been assigned with the same temporal relation by both annotators, 61 event pairs were deemed as causally related. This shows that most of my temporally related regular event pairs are causally related as well.

3.3.1.5 Using VerbOcean Patterns

VerbOcean Chklovski and Pantel [2004a] created lexico-syntactic patterns in order to extract event pairs with various semantic relations from the Web. Specifically, for the temporal relation *happens-before*, VerbOcean used ten patterns such as "to X and then Y", "to X and later Y" and acquired 4,205 event pairs with a temporal "before/after" relation from the Web.

[7] I used the same two annotators that have conducted temporal relation annotations. For this task, the annotator inter-agreement is 0.82 in kappa.

	0 (Seeds)	1	2	3	4	Total
Full System	112	179	271	95	–	657

Table 3.4: Bootstrapping Using VerbOcean Patterns. Reprinted with permission from Yao et al. [2017].

Therefore, I replace my two straightforward temporal relation patterns, EV_A *after (before)* EV_B, with the ten patterns proposed by VerbOcean and use these patterns to acquire seed regular event pairs. However, with exactly the same settings and frequency threshold I used in seed identification, I can only identify seven seed regular event pairs using the same complete Gigaword corpus. In order to obtain more seed event pairs, I lowered the frequency threshold of seeing an event pair in patterns from ten to three. In this case as shown in table 3.4, I obtained 112 seed event pairs, which is still much less than 2110 event pairs that I have acquired. Then with the initial 112 seed regular event pairs, around 500 new event pairs were later learned using exactly the same bootstrapping learning settings I have used. In total, only 657 event pairs were learned by using VerbOcean patterns. Note that the Gigaword corpus I used is much smaller in volume than the Web. Therefore, I hypothesize that VerbOcean patterns are too specific to be productive in identifying regular event pairs from a limited text corpus.

In addition, I compared my learned 4,401 regular event pairs with the 4,205 verb pairs in the *happens-before* relation acquired by VerbOcean[8]. Interestingly, among these two sets, only eight event pairs are the same. This shows that my bootstrapping learning approach recognizes diverse sentential contexts and learns a dramatically different set of temporally related event pairs, compared with VerbOcean which mainly uses specific lexico-syntactic patterns to query the giant Web.

[8]Because event pairs in VerbOcean do not contain arguments, I removed event arguments from my event pairs for direct comparisons.

3.3.2 Weakly Supervised Contextual Temporal Relation Classifier

3.3.2.1 Accuracy of the Classifier

Recall that the contextual temporal relation classifier was trained on the New York Times section of Gigaword. In order to evaluate the accuracy of the classifier, I applied the weakly supervised learned classifier (the full system) to sentential contexts between pairs of events extracted from the Associated Press Worldstream section of Gigaword. I randomly sampled 100 instances from the ones that were labeled by the classifier as indicating a *after* or *before* relation and with a confidence score greater than 0.8. Then for each instance and its pair of events, I asked two annotators to judge whether the sentence indeed describes a *after (before)* temporal relation between the two events. According to the annotations[9], the classifier predicted the correct temporal relation 74% of time.

3.3.2.2 Evaluation Using a Benchmark Dataset

To facilitate direct comparisons, I evaluate both my weakly supervised trained classifier and two supervised trained systems using a benchmark evaluation dataset, the TempEval-3-platinum corpus, which contains 20 news articles annotated with several temporal relations between events. I only evaluate system performance on identifying temporal "before/after" relations.

I compare with two feature-rich supervised trained systems. ClearTK [Bethard, 2013] uses event attributes such as tense, aspect and class, dependency paths and words between two events as features in identifying temporal relations between events. More recently, [Mirza and Tonelli, 2014b] proposes even more sophisticated features including various lexical, grammatical and syntactic features, event durations, temporal signals and temporal discourse connectives etc. In contrast, my neural net based temporal relation classifier is simpler and does not require feature engineering.

Table 3.5 shows the comparison results between these three systems. Note that I ran the original ClearTK system and I re-implemented the system described in [Mirza and Tonelli, 2014b]. In addition, both supervised systems were trained using TimeBank v1.2 [Pustejovsky et al., 2006].

[9]The two annotators achieved a Kappa inter-agreement score of 0.71.

	Approaches	F1	P	R
	ClearTK [Bethard, 2013]	0.27	0.36	0.22
	Mirza and Tonelli [2014b]	0.29	0.24	0.38
	My classifier	0.28	0.35	0.24

Table 3.5: Performance on TempEval-3 Test Data. Reprinted with permission from Yao et al. [2017].

The performance across the three systems is overall low, one reason is that the pairs of events that are in a temporal relation were not provided to the classifiers. Therefore, the classifiers had to identify temporally related event pairs as well as classify their temporal relations. I can see that the weakly supervised classifier achieved roughly equal performance as ClearTK, while the other supervised system presents a a different precision-recall tradeoff. Overall, without using any annotated data or sophisticated hand crafted features, my weakly supervised system achieved a F1-score comparable to both supervised trained systems.

3.4 Conclusion

I presented a weakly supervised bootstrapping approach that learns both regular event pairs and a contextual temporal relation classifier, by exploring the observation that regular event pairs tend to show a consistent temporal relation despite of their diverse contexts. Evaluation shows that the learned regular event pairs are of high quality and rich in commonsense knowledge and domain knowledge. In addition, the weakly supervised trained temporal relation classifier achieves comparable performance with state-of-the-art supervised classifiers.

4. TEMPORAL EVENT KNOWLEDGE ACQUISITION VIA IDENTIFYING NARRATIVES[1]

The previous approach acquires temporal "before/after" knowledge by training an intra-sentence contextual classifier. However, event temporal relations can also be described throughout several sentences. Extracting event temporal knowledge by leveraging specific discourse structures can significantly supplement previous temporal knowledge. According to the narratology theory, a narrative is a discourse presenting a sequence of events arranged in their time order (the plot) and involving specific characters (the characters). Therefore, my second weakly-supervised approach focuses on event temporal "before/after" knowledge acquisition by identifying narratives. More precisely, while a large narrative may contain nonlinear narration such as flashbacks or flash-forwards, this work tries to acquire temporal knowledge by identifying particular story paragraphs that satisfy double temporality from large narratives. The double temporality states that some narrative texts describe a sequence of events following the chronological order and therefore, the temporal order of events matches with their textual order [Walsh, 2001, Riedl and Young, 2010, Grabes, 2013]. For convenience, I will still use narrative identification in the following sections.

This weakly supervised method is designed to capture key elements of narratives in each of two stages. As shown in figure 4.1, in the first stage, I identify the initial batch of narrative paragraphs that satisfy strict rules and the key principles of narratives. Then in the second stage, I train a statistical classifier using the initially identified seed narrative texts and a collection of soft features for capturing the same key principles and other textual devices of narratives. Next, the classifier is applied to identify new narratives from raw texts again. The newly identified narratives will be used to augment seed narratives and the bootstrapping learning process iterates until no enough (specifically, less than 2,000) new narratives can be found. Here, in order to specialize the statistical classifier to each genre, I conduct the learning process on news, novels and blogs separately.

[1]Reprinted with permission from "Temporal Event Knowledge Acquisition via Identifying Narratives" by Wenlin Yao and Ruihong Huang, 2018. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp.537-547

4.1 Key Elements of Narratives

It is generally agreed in narratology [Forster, 1962, Mani, 2012, Pentland, 1999, Bal, 2009] that a narrative presents a sequence of events arranged in their time order (the plot) and involving specific characters (the characters).

Plot. The plot consists of a sequence of closely related events. According to [Bal, 2009], an event in a narrative often describes a "transition from one state to another state, caused or experienced by actors". Moreover, as Mani [2012] illustrates, a narrative is often "an account of past events in someone's life or in the development of something". These prior studies suggest that sentences containing a plot event are likely to have the actantial syntax "NP VP"[2] [Greimas, 1971] with the main verb in the past tense.

Character. A narrative usually describes events caused or experienced by actors. Therefore, a narrative story often has one or two main characters, called protagonists, who are involved in multiple events and tie events together. The main character can be a person or an organization.

Other Textual Devices. A narrative may contain peripheral contents other than events and characters, including time, place, the emotional and psychological states of characters etc., which do not advance the plot but provide essential information to the interpretation of the events [Pentland, 1999]. I use rich Linguistic Inquiry and Word Count (LIWC) [Pennebaker et al., 2015] features to capture a variety of textual devices used to describe such contents.

4.2 Phase One: Weakly Supervised Narrative Identification

In order to acquire rich temporal event knowledge, I first develop a weakly supervised approach that can quickly adapt to identify narrative paragraphs from various text sources.

4.2.1 Rules for Identifying Seed Narratives

Grammar Rules for Identifying Plot Events. Guided by the prior narratology studies [Greimas, 1971, Mani, 2012] and my observations, I use context-free grammar production rules to identify sentences that describe an event in an actantial syntax structure. Specifically, I use three sets of

[2]NP is Noun Phrase and VP is Verb Phrase.

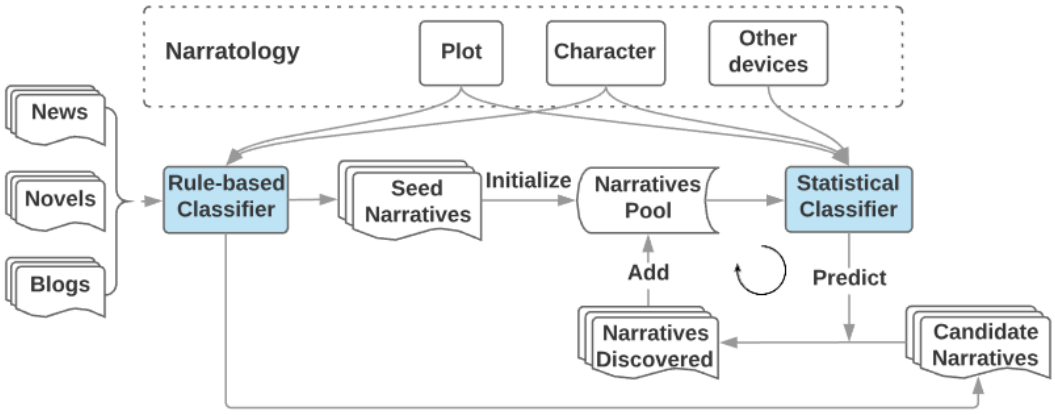

Figure 4.1: Overview of the Narrative Learning System. Reprinted with permission from Yao and Huang [2018].

grammar rules to specify the overall syntactic structure of a sentence. First, I require a sentence to have the basic active voiced structure "S → NP VP" or one of the more complex sentence structures that are derived from the basic structure considering Coordinating Conjunctions (CC), Adverbial Phrase (ADVP) or Prepositional Phrase (PP) attachments[3]. For example, in the narrative of Figure 7.5, the sentence *"Michael Kennedy earned a bachelor's degree from Harvard University in 1980."* has the basic sentence structure "S → NP VP", where the "NP" governs the character mention of 'Michael Kennedy' and the "VP" governs the rest of the sentence and describes a plot event.

In addition, considering that a narrative is usually "an account of past events in someone's life or in the development of something" [Mani, 2012, Dictionary, 2007], I require the headword of the VP to be in the past tense. Furthermore, the subject of the sentence is meant to represent a character. Therefore, I specify 12 grammar rules[4] to require the sentence subject noun phrase to have a simple structure and have a proper noun or pronoun as its head word.

For seed narratives, I consider paragraphs containing at least four sentences and I require 60% or more sentences to satisfy the sentence structure specified above. I also require a narrative paragraph to contain no more than 20% of sentences that are interrogative, exclamatory or dialogue,

[3]I manually identified 14 top-level sentence production rules, for example, "S → NP ADVP VP", "S → PP , NP VP" and "S → S CC S". Appendix shows all the rules.

[4]The example NP rules include "NP → NNP", "NP → NP CC NP" and "NP → DT NNP".

which normally do not contain any plot events. The specific parameter settings are mainly determined based on my observations and analysis of narrative samples. The threshold of 60% for "sentences with actantial structure" was set to reflect the observation that sentences in a narrative paragraph usually (over half) have an actantial structure. A small portion (20%) of interrogative, exclamatory or dialogue sentences is allowed to reflect the observation that many paragraphs are overall narratives even though they may contain 1 or 2 such sentences, so that I achieve a good coverage in narrative identification.

The Character Rule. A narrative usually has a protagonist character that appears in multiple sentences and ties a sequence of events, therefore, I also specify a rule requiring a narrative paragraph to have a protagonist character. Concretely, inspired by Eisenberg and Finlayson [2017], I applied the named entity recognizer [Finkel et al., 2005] and entity coreference resolver [Lee et al., 2013] from the CoreNLP toolkit [Manning et al., 2014b] to identify the longest entity chain in a paragraph that has at least one mention recognized as a *Person* or *Organization*, or a gendered pronoun. Then I calculate the normalized length of this entity chain by dividing the number of entity mentions by the number of sentences in the paragraph. I require the normalized length of this longest entity chain to be ≥ 0.4, meaning that 40% or more sentences in a narrative mention a character [5].

4.2.2 The Statistical Classifier for Identifying New Narratives

Using the seed narrative paragraphs identified in the first stage as positive instances, I train a statistical classifier to continue to identify more narrative paragraphs that may not satisfy the specific rules. I also prepare negative instances to compete with positive narrative paragraphs in training. Negative instances are paragraphs that are not likely to be narratives and do not present a plot or protagonist character, but are similar to seed narratives in others aspects. Specifically, similar to seed narratives, I require a non-narrative paragraph to contain at least four sentences with no more than 20% of sentences being interrogative, exclamatory or dialogue; but in contrast to seed narratives, a non-narrative paragraph should contain 30% of or fewer sentences that have

[5] 40% was chosen to reflect that a narrative paragraph often contains a main character that is commonly mentioned across sentences (half or a bit less than half of all the sentences).

the actantial sentence structure, where the longest character entity chain should not span over 20% of sentences. I randomly sample such non-narrative paragraphs that are five times of narrative paragraphs [6].

In addition, since it is infeasible to apply the trained classifier to all the paragraphs in a large text corpus, such as the Gigaword corpus [Graff and Cieri, 2003], I identify candidate narrative paragraphs and only apply the statistical classifier to these candidate paragraphs. Specifically, I require a candidate paragraph to satisfy all the constraints used for identifying seed narrative paragraphs but contain only 30%[7] or more sentences with an actantial structure and have the longest character entity chain spanning over 20%[8] of or more sentences.

I choose Maximum Entropy [Berger et al., 1996] as the classifier. Specifically, I use the MaxEnt model implementation in the LIBLINEAR library[9] [Fan et al., 2008] with default parameter settings. Next, I describe the features used to capture the key elements of narratives.

Features for Identifying Plot Events: Realizing that grammar production rules are effective in identifying sentences that contain a plot event, I encode all the production rules as features in the statistical classifier.

Specifically, for each narrative paragraph, I use the frequency of all syntactic production rules as features. Note that the bottom level syntactic production rules have the form of POS tag → WORD and contain a lexical word, which made these rules dependent on specific contexts of a paragraph. Therefore, I exclude these bottom level production rules from the feature set in order to model generalizable narrative elements rather than specific contents of a paragraph.

In addition, to capture potential event sequence overlaps between new narratives and the already learned narratives, I build a verb bigram language model using verb sequences extracted from the learned narrative paragraphs and calculate the perplexity score (as a feature) of the verb sequence in a candidate narrative paragraph. Specifically, I calculate the perplexity score of an

[6] I used the skewed pos:neg ratio of 1:5 in all bootstrapping iterations to reflect the observation that there are generally many more non-narrative paragraphs than narrative paragraphs in a document.

[7] This value is half of the corresponding thresshold used for identifying seed narrative paragraphs.

[8] This value is half of the corresponding thresshold used for identifying seed narrative paragraphs.

[9] `https://www.csie.ntu.edu.tw/~cjlin/liblinear/`

event sequence that is normalized by the number of events, $PP(e_1, ..., e_N) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(e_i|e_{i-1})}}$, where N is the total number of events in a sequence and e_i is a event word. I approximate $P(e_i|e_{i-1}) = \frac{C(e_{i-1},e_i)}{C(e_{i-1})}$, where $C(e_{i-1})$ is the number of occurrences of e_{i-1} and $C(e_{i-1}, e_i)$ is the number of co-occurrences of e_{i-1} and e_i. $C(e_{i-1}, e_i)$ and $C(e_{i-1})$ are calculated based on all event sequences from known narrative paragraphs.

Features for the Protagonist Characters: I consider the longest three coreferent entity chains in a paragraph that have at least one mention recognized as a *Person* or *Organization*, or a gendered pronoun. Similar to the seed narrative identification stage, I obtain the normalized length of each entity chain by dividing the number of entity mentions with the number of sentences in the paragraph. In addition, I also observe that a protagonist character appears frequently in the surrounding paragraphs as well, therefore, I calculate the normalized length of each entity chain based on its presences in the target paragraph as well as one preceding paragraph and one following paragraph. I use 6 normalized lengths (3 from the target paragraph [10] and 3 from surrounding paragraphs) as features.

Other Writing Style Features: I create a feature for each semantic category in the Linguistic Inquiry and Word Count (LIWC) dictionary [Pennebaker et al., 2015], and the feature value is the total number of occurrences of all words in that category. These LIWC features capture presences of certain types of words, such as words denoting relativity (e.g., motion, time, space) and words referring to psychological processes (e.g., emotion and cognitive). In addition, I encode Parts-of-Speech (POS) tag frequencies as features as well which have been shown effective in identifying text genres and writing styles.

4.2.3 Identifying Narrative Paragraphs from Three Text Corpora

My weakly supervised system is based on the principles shared across all narratives, so it can be applied to different text sources for identifying narratives. I considered three types of texts: (1) **News Articles**. News articles contain narrative paragraphs to describe the background

[10] Specifically, the lengths of the longest, second longest and third longest entity chains.

	0 (Seeds)	1	2	3	4	Total
News	20k	40k	12k	5k	1k	78k
Novels	75k	82k	24k	6k	2k	189k
Blogs	6k	10k	3k	1k	-	20k
Sum	101k	132k	39k	12k	3k	287k

Table 4.1: Number of new narratives generated after each bootstrapping iteration. Reprinted with permission from Yao and Huang [2018].

of an important figure or to provide details for a significant event. I use English Gigaword 5th edition [Graff and Cieri, 2003, Napoles et al., 2012], which contains 10 million news articles. (2) **Novel Books**. Novels contain rich narratives to describe actions by characters. BookCorpus [Zhu et al., 2015] is a large collection of free novel books written by unpublished authors, which contains 11,038 books of 16 different sub-genres (e.g., Romance, Historical, Adventure, etc.). (3) **Blogs**. Vast publicly accessible blogs also contain narratives because "personal life and experiences" is a primary topic of blog posts [Lenhart, 2006]. I use the Blog Authorship Corpus [Schler et al., 2006] collected from the blogger.com website, which consists of 680k posts written by thousands of authors. I applied the Stanford CoreNLP tools [Manning et al., 2014b] to the three text corpora to obtain POS tags, parse trees, named entities, coreference chains, etc.

In order to combat semantic drifts [McIntosh and Curran, 2009] in bootstrapping learning, I set the initial selection confidence score produced by the statistical classifier at 0.5 and increase it by 0.05 after each iteration. The bootstrapping system runs for four iterations and learns 287k narrative paragraphs in total. Table 4.1 shows the number of narratives that were obtained in the seeding stage and in each bootstrapping iteration from each text corpus.

4.3 Phase Two: Extract Event Temporal Knowledge from Narratives

Narratives I obtained from the first phase may describe specific stories and contain uncommon events or event transitions. Therefore, I apply Pointwise Mutual Information (PMI) based statistical metrics to measure strengths of event temporal relations in order to identify general knowledge that is not specific to any particular story. My goal is to learn event pairs and longer event chains with

events completely ordered in the temporal "before/after" relation.

First, by leveraging the double temporality characteristic of narratives, I only consider event pairs and longer event chains with 3-5 events that have occurred as a segment in at least one event sequence extracted from a narrative paragraph. Specifically, I extract the event sequence (the plot) from a narrative paragraph by finding the main event in each sentence and chaining the main events[11] according to their textual order.

Then I rank candidate event pairs based on two factors, how strongly associated two events are and how common they appear in a particular temporal order. I adopt the existing metric, Causal Potential (CP), which has been applied to acquire causally related events [Beamer and Girju, 2009] and exactly measures the two aspects. Specifically, the CP score of an event pair is calculated using the following equation:

$$cp(e_i, e_j) = pmi(e_i, e_j) + log\frac{P(e_i \rightarrow e_j)}{P(e_j \rightarrow e_i)} \tag{4.1}$$

where, the first part refers to the Pointwise Mutual Information (PMI) between two events and the second part measures the relative ordering or two events. $P(e_i \rightarrow e_j)$ refers to the probability that e_i occurs before e_j in a text, which is proportional to the raw frequency of the pair. PMI measures the association strength of two events, formally, $pmi(e_i, e_j) = log\frac{P(e_i,e_j)}{P(e_i)P(e_j)}$, $P(e_i) = \frac{C(e_i)}{\sum_x C(e_x)}$ and $P(e_i, e_j) = \frac{C(e_i,e_j)}{\sum_x \sum_y C(e_x,e_y)}$, where, x and y refer to all the events in a corpus, $C(e_i)$ is the number of occurrences of e_i, $C(e_i, e_j)$ is the number of co-occurrences of e_i and e_j.

While each candidate pair of events should have appeared consecutively as a segment in at least one narrative paragraph, when calculating the CP score, I consider event co-occurrences even when two events are not consecutive in a narrative paragraph but have one or two other events in between. Specifically, the same as in [Hu and Walker, 2017], I calculate separate CP scores based on event co-occurrences with zero (consecutive), one or two events in between, and use the weighted average CP score for ranking an event pair, formally, $CP(e_i, e_j) = \sum_{d=1}^{3} \frac{cp_d(e_i,e_j)}{d}$.

Then I rank longer event sequences based on CP scores for individual event pairs that are included in an event sequence. However, an event sequence of length n is more than $n-1$ event

[11] I only consider main events that are in base verb forms or in the past tense, by requiring their POS tags to be VB, VBP, VBZ or VBD.

pairs with any two consecutive events as a pair. I prefer event sequences that are coherent overall, where the events that are one or two events away are highly related as well. Therefore, I define the following metric to measure the quality of an event sequence:

$$CP(e_1, e_2, \cdots, e_n) = \frac{\sum_{d=1}^{3} \sum_{j=1}^{n-d} \frac{CP(e_j, e_{j+d})}{d}}{n-1}. \tag{4.2}$$

4.4 Evaluation

4.4.1 Precision of Narrative Paragraphs

From all the learned narrative paragraphs, I randomly selected 150 texts, with 25 texts selected from narratives learned in each of the two stages (i.e., seed narratives and bootstrapped narratives) using each of the three text corpora (i.e., news, novels, and blogs). Following the same definition "A story is a narrative of events arranged in their time sequence" [Forster, 1962, Gordon and Swanson, 2009], two human adjudicators were asked to judge whether each text is a narrative or a non-narrative. In order to obtain high inter-agreements, before the official annotations, I trained the two annotators for several iterations. Note that the texts I used in training annotators are different from the final texts I used for evaluation purposes. The overall kappa inter-agreement between the two annotators is 0.77.

Table 4.2 shows the precision of narratives learned in the two stages using the three corpora. I determined that a text is a correct narrative if both annotators labeled it as a narrative. We can see that on average, the rule-based classifier achieves the precision of 88% on initializing seed narratives and the statistical classifier achieves the precision of 84% on bootstrapping new ones. Using narratology based features enables the statistical classifier to extensively learn new narrative, and meanwhile maintain a high precision.

4.4.2 Precision of Event Pairs and Chains

To evaluate the quality of the extracted event pairs and chains, I randomly sampled 20 pairs (2%) from every 1,000 event pairs up to the top 18,929 pairs with CP score $\geq$ 2.0 (380 pairs

Narratives	Seed	Bootstrapped
News	0.84	0.72
Novel	0.88	0.92
Blogs	0.92	0.88
AVG	0.88	0.84

Table 4.2: Precision of narratives based on human annotation. Reprinted with permission from Yao and Huang [2018].

pairs	graduate → teach (5.7), meet → marry (5.3) pick up → carry (6.3), park → get out (7.3) turn around → face (6.5), dial → ring (6.3)
chains	drive → park → get out (7.8) toss → fly → land (5.9) grow up → attend → graduate → marry (6.9) contact → call → invite → accept (4.2) knock → open → reach → pull out → hold (6.0)

Table 4.3: Examples of event pairs and chains (with CP scores). → represents *before* relation. Reprinted with permission from Yao and Huang [2018].

selected in total), and 10 chains (1%) from every 1,000 up to the top 25,000 event chains[12] (250 chains selected in total). The average CP scores for all event pairs and all event chains I considered are 2.9 and 5.1 respectively. Two human adjudicators were asked to judge whether or not events are likely to occur in the temporal order shown. For event chains, I have one additional criterion requiring that events form a coherent sequence overall. An event pair/chain is deemed correct if both annotators labeled it as correct. The two annotators achieved kappa inter-agreement scores of 0.71 and 0.66, on annotating event pairs and event chains respectively.

As we know, coverage on acquired knowledge is often hard to evaluate because I do not have a complete knowledge base to compare to. Thus, I propose a pseudo recall metric to evaluate the coverage of event knowledge I acquired. Regneri et al. [2010] collected Event Sequence Descriptions (ESDs) of several types of human activities (e.g., baking a cake, going to the theater, etc.) using

[12]It turns out that many event chains have a high CP score close to 5.0, so I decided not to use a cut-off CP score of event chains but simply chose to evaluate the top 25,000 event chains.

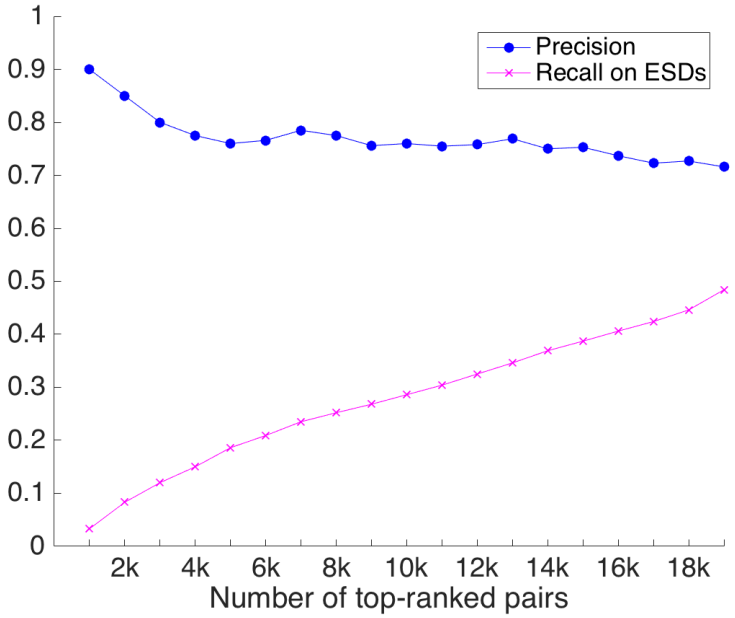

Figure 4.2: Top-ranked event pairs evaluation. Reprinted with permission from Yao and Huang [2018].

# of top chains	5k	10k	15k	20k	25k
Precision	0.76	0.8	0.75	0.73	0.69

Table 4.4: Precision of top-ranked event chains. Reprinted with permission from Yao and Huang [2018].

crowdsourcing. My first pseudo recall score is calculated based on how many consecutive event pairs in human-written scripts can be found in my top-ranked event pairs. Figure 4.2 illustrates the precision of top-ranked pairs based on human annotation and the pseudo recall score based on ESDs. We can see that about 75% of the top 19k event pairs are correct, which captures 48% of human-written script knowledge in ESDs. In addition, table 4.4 shows the precision of top-ranked event chains with 3 to 5 events. Among the top 25k event chains, about 70% are correctly ordered with the temporal "after" relation. Table 4.3 shows several examples of event pairs and chains.

Models	Acc.(%)
Choubey and Huang [2017]	51.2
+ CP score	**52.3**

Table 4.5: Results on TimeBank corpus. Reprinted with permission from Yao and Huang [2018].

4.4.3 Improving Temporal Relation Classification by Incorporating Event Knowledge

To find out whether the learned temporal event knowledge can help with improving temporal relation classification performance, I conducted experiments on a benchmark dataset - TimeBank corpus v1.2, which contains 2308 event pairs that are annotated with 14 temporal relations [13].

To facilitate direct comparisons, I used the same state-of-the-art temporal relation classification system as described in my previous work Choubey and Huang [2017] and considered all the 14 relations in classification. Choubey and Huang [2017] forms three sequences (i.e., word forms, POS tags, and dependency relations) of context words that align with the dependency path between two event mentions and uses three bi-directional LSTMs to get the embedding of each sequence. The final fully connected layer maps the concatenated embeddings of all sequences to 14 fine-grained temporal relations. I applied the same model here, but if an event pair appears in my learned list of event pairs, I concatenated the CP score of the event pair as additional evidence in the final layer. To be consistent with Choubey and Huang [2017], I used the same train/test splitting, the same parameters for the neural network and only considered intra-sentence event pairs. Table 4.5 shows that by incorporating my learned event knowledge, the overall prediction accuracy was improved by 1.1%. Not surprisingly, out of the 14 temporal relations, the performance on the relation *before* was improved the most by 4.9%.

4.4.4 Narrative Cloze

Multiple Choice version of the Narrative Cloze task (MCNC) proposed by Granroth-Wilding and Clark [2016], Wang et al. [2017b], aims to evaluate understanding of a script by predicting the

[13] Specifically, the 14 relations are *simultaneous, before, after, ibefore, iafter, begins, begun by, ends, ended by, includes, is included, during, during inv, identity*

Method	Acc.(%)
[Chambers and Jurafsky, 2008]	30.92
[Granroth-Wilding and Clark, 2016]	43.28
[Pichotta and Mooney, 2016]	43.17
[Wang et al., 2017b]	46.67
My Results	**48.83**

Table 4.6: Results on MCNC task. Reprinted with permission from Yao and Huang [2018].

next event given several context events. Presenting a chain of contextual events $e_1, e_2, ..., e_{n-1}$, the task is to select the next event from five event candidates, one of which is correct and the others are randomly sampled elsewhere in the corpus. Following the same settings of Wang et al. [2017b] and Granroth-Wilding and Clark [2016], I adapted the dataset (test set) of Chambers and Jurafsky [2008] to the multiple choice setting. The dataset contains 69 documents and 349 multiple choice questions.

I calculated a PMI score between a candidate event and each context event $e_1, e_2, ..., e_{n-1}$ based on event sequences extracted from my learned 287k narratives and I chose the event that have the highest sum score of all individual PMI scores. Since the prediction accuracy on 349 multiple choice questions depends on the random initialization of four negative candidate events, I ran the experiment 10 times and took the average accuracy as the final performance.

Table 4.6 shows the comparisons of my results with the performance of several previous models, which were all trained with 1,500k event chains extracted from the NYT portion of the Gigaword corpus [Graff and Cieri, 2003]. Each event chain consists of a sequence of verbs sharing an actor within a news article. Except Chambers and Jurafsky [2008], other recent models utilized more and more sophisticated neural language models. Granroth-Wilding and Clark [2016] proposed a two layer neural network model that learns embeddings of event predicates and their arguments for predicting the next event. Pichotta and Mooney [2016] introduced a LSTM-based language model for event prediction. Wang et al. [2017b] used dynamic memory as attention in LSTM for prediction. It is encouraging that by using event knowledge extracted from automatically identified narratives, I achieved the best event prediction performance, which is 2.2% higher

than the best neural network model.

4.5 Conclusion

This section presents a novel approach for leveraging the double temporality characteristic of narrative texts and acquiring temporal event knowledge across sentences in narrative paragraphs. I developed a weakly supervised system that explores narratology principles and identifies narrative texts from three text corpora of distinct genres. The temporal event knowledge distilled from narrative texts were shown useful to improve temporal relation classification and outperform several neural language models on the narrative cloze task. For the future work, I plan to expand event temporal knowledge acquisition by dealing with event sense disambiguation and event synonym identification (e.g., drag, pull and haul).

5. WEAKLY SUPERVISED SUBEVENT KNOWLEDGE ACQUISITION[1]

Previous two models mainly focuses on acquiring event temporal knowledge. Another event relation - the subevent-parentevent relation also widely exist between events. The subevent-parentevent relation indicates that several events (subevents) happen as parts of another event (parentevent) spatio-temporally. Subevent knowledge is useful for discourse analysis and event-centric applications. Acknowledging the scarcity of subevent knowledge, I propose a weakly supervised approach to extract subevent relation tuples from text and build the first large scale subevent knowledge base.

Figure 5.1 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 5.1). I 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, I 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, I find occurrences of seed pairs in a large text corpus to quickly generate many subevent relation instances, I will also create negative instances to train the subevent relation classifier (Section 5.2). 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 5.3). I use the English Gigaword [Napoles et al., 2012] as the text corpus.

5.1 Weak Supervision

5.1.1 Seed Event Pair Identification

I use six preposition patterns (i.e., *during, in, amid, throughout, including*, and *within*) to extract candidate seed event pairs. Specifically, I use dependency relations[2] to recognize preposition

[1]Reprinted with permission from "Weakly Supervised Subevent Knowledge Acquisition" by Wenlin Yao, Zeyu Dai, Maitreyi Ramaswamy, Bonan Min, Ruihong Huang, 2020. In Proceedings of The 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.5345-5356.

[2]I use Stanford dependency relations [Manning et al., 2014a], e.g., *prep_during.*

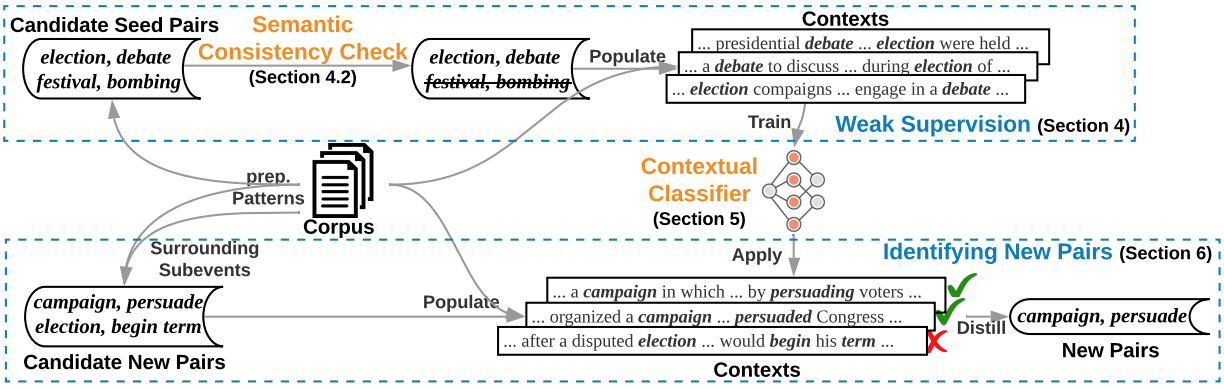

Figure 5.1: Overview of the Subevent Knowledge Acquisition System. Reprinted with permission from Yao et al. [2020a].

patterns, and extract the governor word and dependent word of each pattern. I 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* v.s *sign*, or, *attack on troops* v.s *attack*. I consider both verb event phrases and noun event phrases (Appendix 2.1 provides more details). I 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, I do not consider event pairs in which neither event has an argument.

To select seed subevent pairs, I consider event pairs that co-occur with at least two different patterns for at least three times. In this way, I 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, I 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.

5.1.2 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, I 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 I cluster event phrases into groups so that any two event phrases within a group are semantically compatible. Therefore, when the clustering operation is completed, I will recognize an event pair as a spurious subevent relation pair if its two events fall into different clusters. Next, I describe details on graph construction and the clustering algorithm I used.

Graph Construction: Given a set of event pairs needing the semantic consistency check, I construct an undirected graph $G(V, E)$, where each node in V represents a unique event phrase. I connect event phrases with two types of weighted edges. First, for each candidate subevent relation tuple, I create an edge of weight 1.0 between the parent event and the child event. Second, I create an edge between any two event phrases if their similarity[3] is greater than a certain threshold[4], and the edge weight is their similarity score. If two event phrases are already connected because they are a candidate subevent relation pair, I add their similarity score to the edge weight.

Next, I 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, I create a new node for its event word representing the event word definition. If the event word has multiple meanings and therefore multiple definitions, I 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, I connect each definition node of a parent event with its children events, if their similarity[5] is over the same simi-

[3]To calculate the similarity between two event phrases, I pair each word from one event phrase (either the event word or an argument) with each word from the other event phrase and calculate the similarity between two word embeddings, then the similarity between two event phrases is the average of their word pair similarities. I used word2vec word embeddings.

[4]I set the similarity threshold as 0.3, after inspecting 200 randomly selected event pairs with their similarities.

[5]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.

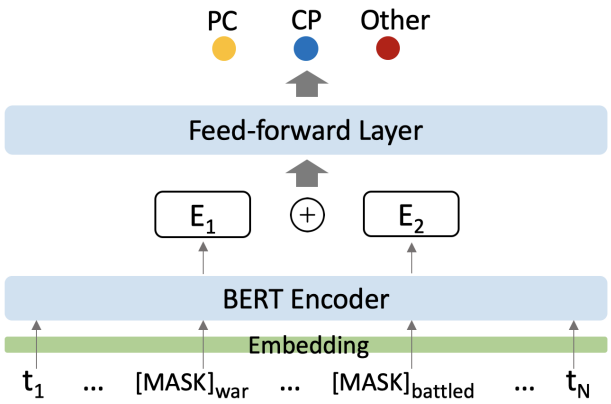

Figure 5.2: BERT-based Contextual Classifier

larity threshold used previously.

The Clustering Algorithm: I 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.

After performing the semantic consistency check[6], I retained around 30K seed event pairs. I 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.

5.2 The Contextual Classifier Using BERT

Recently, BERT [Devlin et al., 2019] pretrained on massive data has achieved high performance on various NLP tasks. I fine-tune a pretrained BERT model to build the contextual classifier for

[6]Event clusters often become stable soon after 50 iterations, to ensure convergence, I ran the algorithm for 60 iterations.

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

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, I 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. Figure 5.2 shows the overall structure of the classifier. 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*, I 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 my experiments, I 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]. I 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.

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. I include two types of negative instances to fine-tune the BERT classifier.

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

First, I randomly sample sentences that contain an event pair different from any seed pair or candidate pair (Section 5.3.1) as negative instances. I 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, I 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, I also include sentences containing temporally or causally related events as negative instances. Specifically, I apply a similar strategy - using patterns[9] to extract temporal and causal event pairs and then search for these pairs 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. I 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, I obtained around 1.8 million negative training instances.

5.3 Identifying New Subevent Pairs

I 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, I first identify candidate event pairs that are likely to have the subevent relation. Then, I apply the contextual classifier to examine instances of each candidate event pair in order to determine valid subevent relation pairs.

5.3.1 Candidate Event Pairs

I 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, I consider event pairs that co-occur with any pattern for at least two times as candidate event pairs. In this way, I identified 1.4 million candidate event pairs from the

[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.

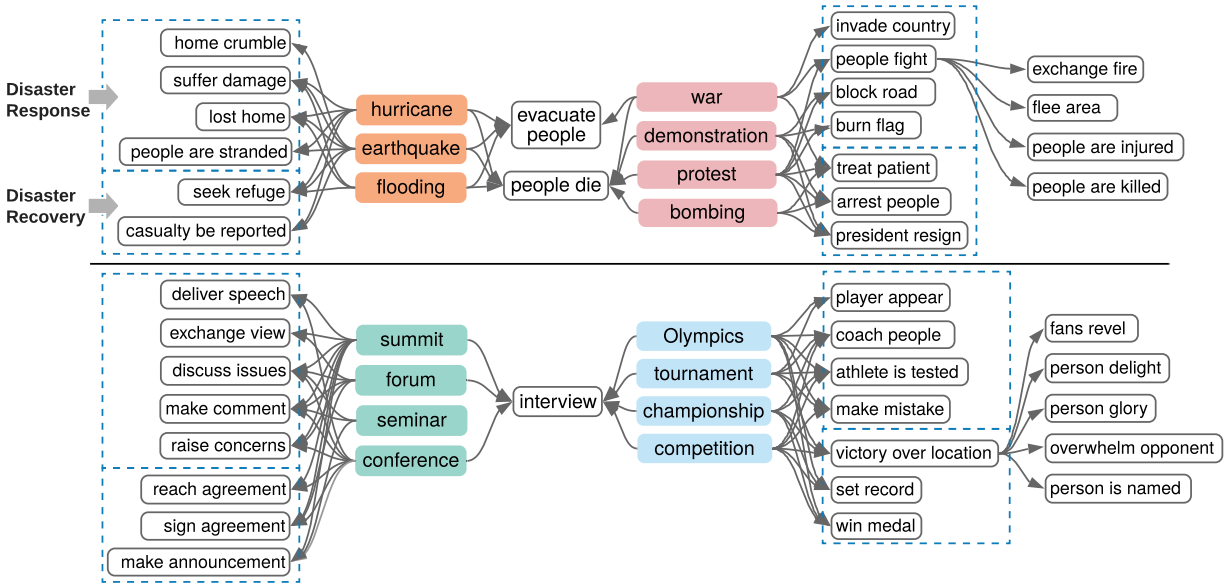

Figure 5.3: Example Subevent Knowledge Graph (→ denotes the Parent→Child subevent relation). Four colors indicate four groups of parent events where parent events in the same group commonly share children events. Children events circled by the same blue dash box describe a stage of development of parent events. Reprinted with permission from Yao et al. [2020a].

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, I 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, I observe that the additional subevents often share the subject or direct object with the subevent of the seed tuple, as a consequence, I 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, I extracted around 89K candidate event pairs from the Gigaword corpus.

5.3.2 New Subevent Pair Selection Criteria

I 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, I disregard rare event pairs and require that

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

Table 5.1: Performance of the Contextual Classifier. Reprinted with permission from Yao et al. [2020a].

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.

5.3.3 Example Subevent Knowledge Graph

The initial exploration of the learned subevent knowledge shows two interesting observations of event hierarchies. Figure 5.3 shows an example event graph. First, I 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, I 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.

5.4 Intrinsic and Extrinsic Evaluations

5.4.1 Precision of the Contextual Classifier

The contextual classifier is a key component of my learning approach. I evaluate the performance of the BERT contextual classifier on identifying subevent relations against all the 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

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 5.2: Subevent Relation Identification. P/R/F1 (%). I predict Parent-Child and Child-Parent subevent relations and report the micro-average performance. Reprinted with permission from Yao et al. [2020a].

relations. Since the contextual classifier mainly performs at the sentence level, I only consider to identify intra-sentence subevent relations in the RED dataset[10]. Table 5.1 shows the comparisons between two training settings - the BERT classifier trained on seed pairs before v.s 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. Overall, without using any annotated data, the classifier achieves the precision of 56%[11].

5.4.2 Accuracy of Acquired Subevent Pairs

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

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

[11]While the precision is not perfect, note that I only retain a candidate subevent relation pair if the majority (3+) of its sentential contexts show the subevent relation.

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 5.3: Multi-class Classification Results on the PDTB dataset. I report accuracy (Acc), macro-average (Macro) P/R/F1 (%) over four implicit discourse relation categories as well as performance on each category. Reprinted with permission from Yao et al. [2020a].

5.4.3 Coverage of Acquired Subevent Pairs

To see whether the acquired subevent knowledge has good coverage of diverse event types, I 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. I found that 73.8% (656/889) of events in ACE and 66.9% (934/1396) of events in KBP match[12] with events in the acquired subevent pairs.

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

5.4.4 Subevent Relation Identification

To find out whether the learned subevent knowledge can be used to improve subevent relation identification in text, I conducted experiments on two datasets, RED[13] and HiEve[14] [Glavaš et al., 2014]. In my experiments, I consider intra-sentence and cross-sentence event pairs separately. I randomly split data into five folds and conduct cross-validation for evaluation. I fine-tune the same

[12]I ignore event arguments and only consider to match event word lemmas, because I aim to evaluate the coverage on general event types instead of specific events.

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

[14]HiEve annotated 3,200 event mentions and their subevents as well as coreference relations in 100 documents. I 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	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 5.4: Temporal Relation Identification. P/R/F1 (%). I predict Before and After temporal relations and report the micro-average performance. Reprinted with permission from Yao et al. [2020a].

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 5.5: Causal Relation Identification. P/R/F1 (%). I predict Cause-Effect and Effect-Cause relations and report the micro-average performance. Reprinted with permission from Yao et al. [2020a].

BERT model using RED or HiEve annotations to predict subevent relations vs others[15][16]. Note that for cross-sentence event pairs, I simply concatenate two sentences and insert in between the special token `[SEP]` used in the original BERT.

I 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, I check if they match with my learned

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

[16]For the HiEve dataset, I 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, I generated 3.5K intra-sentence and 59.5K cross-sentence event mention pairs as others.

subevent relation tuples. I 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. I 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, I train a BiLSTM encoder[17] to build event phrase embeddings, using the knowledge representation learning model TransE [Bordes et al., 2013][18] such that $\mathbf{p} + \mathbf{r} \approx \mathbf{c}$ given a parent-child event pair (p, c) having the subevent relation r. I 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, I 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 5.2. 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.

5.4.5 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, I use the same RED[19] dataset plus two more datasets, TimeBank v1.2[20] [Pustejovsky et al., 2003] and Event Storyline Corpus (ESC) v1.5[21] [Caselli and Inel, 2018], dedicated to evaluate temporal relation and causal relation identification systems re-

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

[18]I trained TransE for 20 iterations.

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

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

[21]ESC has 1,649 intra-sentence and 3,952 cross-sentence causal relations. I exhaustively create event mention pairs 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, I generated 4.1K intra-sentence and 34K cross-sentence event mention pairs as others.

spectively. I use the same experimental settings, including 5-fold cross-validations and evaluating predictions of intra- and cross-sentence cases separately. In addition, I repurpose the BERT model to predict temporal relations v.s others or predict causal relations v.s others, and I use the same two methods to incorporate the learned subevent knowledge.

Table 5.4 and 5.5 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.

5.4.6 Implicit Discourse Relation Classification

I 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, I used a recent discourse parsing system proposed by Dai and Huang [2019] which can easily incorporate 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] [22].

Table 5.3 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.

[22] I use the source code provided by the first author of Dai and Huang [2019].

5.5 Conclusions

I 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. I 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, I would like to explore uses of the subevent knowledge base for other event-oriented applications such as event tracking.

6. INCORPORATING EVENT KNOWLEDGE INTO A GRAPH AND LEARNING DISTRIBUTED REPRESENTATIONS OF EVENTS

Understanding the semantics of texts requires comprehensive understanding of different relations among events. In this chapter, I will first discuss learning event causality knowledge. Next, I will talk about constructing a general event knowledge graph based on all acquired knowledge and learning event distributed embeddings.

6.1 Incorporating All Event Knowledge into A General Ontology Graph

Besides temporal and subevent relations, causal relations are also important in NLP applications, particularly event prediction and causal reasoning. The weakly-supervised approach presented in Chapter 5 only relies on weak supervision, where I identify the initial set of an event relation from a big text corpus using linguistic patterns and fine-tune a powerful contextual BERT classifier to extract more knowledge tuples. Therefore, in this chapter, I apply the same approach to acquire event temporal knowledge and causal knowledge. Specifically, I consider three temporal patterns that indicate temporal *happen_before*, *happen_after* relations and seven causal patterns that indicate *cause_effect* relation to identify seed and candidate event pairs. Next, I populate the identified event pairs in the text corpus to collect sentences containing a seed pair, which are used to fine-tune a contextual BERT classifier. After training, BERT classifier is applied to sentences containing candidate pairs to identify new event pairs in temporal or causal relations. In this way, I have built a temporal event knowledge base of 117K event tuples in *after* or *before* relation and 49K event tuples in *cause_effect* relation.

To incorporate event temporal knowledge extracted from 287K narrative paragraphs (Chapter 4) with other event relational knowledge, I adopt the same event representation in Section 2.1 to event extraction by attaching arguments to verb and noun events. Recall that I apply Pointwise Mutual Information (PMI) based metrics to measure strengths of event temporal relations in order to identify general knowledge that is not specific to any particular story. I rank event

Method	Temporal pairs	Subevent pairs	Causal pairs
CNN intra-sentence classifier	4.4K	-	-
Acquision via identifying narratives	57K	-	-
BERT intra-sentence classifier	117K	239K	49K

Table 6.1: Statistics of Event Pairs Acquired by Each Weakly-supervised Approach.

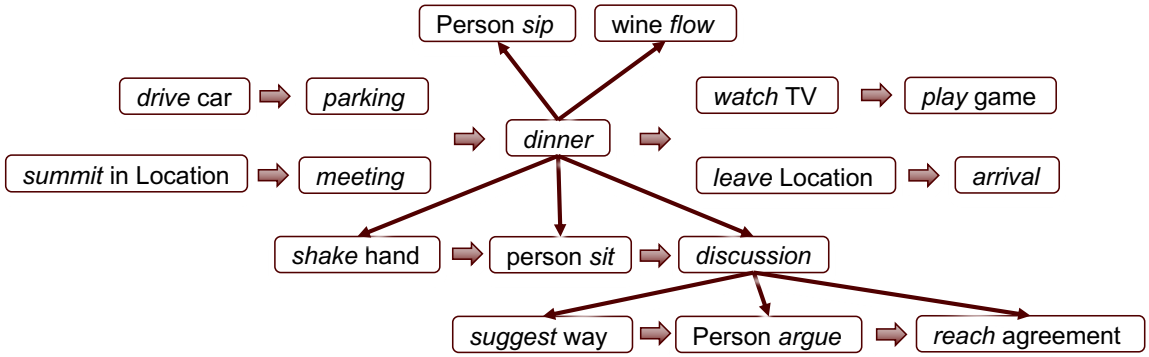

Figure 6.1: Example Constructed Knowledge Graph. ⇨ denotes the temporal *happens_before* relation and → denotes the Parent→Child subevent relation.

pairs extracted from narratives based on the same metric, Causal Potential (CP) - $CP(e_i, e_j) = pmi(e_i, e_j) + log\frac{P(e_i \rightarrow e_j)}{P(e_j \rightarrow e_i)}$. However, attaching event arguments makes event representations more sparse than only considering verb predicates as events, so I apply the window size of 3 to count the number of co-occurrence ($C(e_i, e_j)$) of e_i and e_j when calculating $P(e_i \rightarrow e_j)$. Formally, $pmi(e_i, e_j) = log\frac{P(e_i, e_j)}{P(e_i)P(e_j)}$, $P(e_i) = \frac{C(e_i)}{\sum_x C(e_x)}$ and $P(e_i, e_j) = \frac{C(e_i, e_j)}{\sum_x \sum_y C(e_x, e_y)}$, where, x and y refer to all the events in a corpus, $C(e_i)$ is the number of occurrences of e_i, and $\sum_x C(e_x)$ is the total number of occurrences of all events in the corpus. $C(e_i, e_j)$ is the number of co-occurrences of e_i and e_j (two events occur within the window size 3). In this way, I extracted 57K event pairs with CP score >= 1.0.

Table 6.1 shows the number of event pairs extracted from each weakly-supervised approach. Finally, I construct an event knowledge graph $G(V, E)$ using all acquired event pairs. Formally, V represents all event nodes where each node is an event phrase, and E represents directional edges where each edge belongs to {*happens-before*, *parent-subevent*, *cause-effect*}. The final event

knowledge graph contains 126K event nodes, 174K *happens-before* edges, 239K *parent-subevent* edges and 49K *cause-effect* edges. In the following section, I will use temporal relation, subevent relation, causal relation to refer to three relation types.

Figure 6.1 gives an example of the constructed knowledge graph. There are two chains of events "*drive* car ⇨ *parking* ⇨ *dinner* ⇨ *watch* TV ⇨ *play* game" and "*summit* in Location ⇨ *meeting* ⇨ *dinner* ⇨ *leave* Location ⇨ *arrival*" that happen sequentially share a same event "dinner". The upper event chain describes a script of people's daily life, where the "dinner" event further contains two subevents "Person *sip*" and "wine *flow*" about how people eat during a "dinner" event. The lower event chain describes a script related to a business vacation, where the "dinner" event contains "*shake* hand", "person *sit*" and "*discussion*". Additionally, event "discussion" further contains "*suggest* way", "Person *argue*" and "*reach* agreement" about how people exchange views during the "discussion". We can see the final event knowledge graph can describe human routine activities as a network of event nodes with temporal sequences depicting narrative scripts and event hierarchies elaborating what subevents constitute the root event node. The final knowledge graph can serve as a comprehensive common-sense knowledge base for interpreting complicated world dynamics.

6.2 Learning Distributed Representations of Events

Recently, knowledge graph (KG) embedding that projects entities and relations into continuous vector spaces has been shown to be effective to preserve the inherent structure of the knowledge graph and has been successfully applied to many NLP applications. To investigate how event semantics that is encoded in acquired event knowledge graph is different from word semantics, I adopt the **event knowledge graph embedding** method (Chapter 5) to map each event phrase representation into a vector representation so that their graph structure (i.e., typical neighbor events connected by different edges) is encoded.

Event Knowledge Graph Embedding: Similar to Chapter 5, given an event pair $(e1, r, e2)$, I interpret the relation r as a translation vector $\mathbf{r}$ so that event node $e1$ and $e2$ can be connected by vector $\mathbf{r}$, namely, $\mathbf{p} + \mathbf{r} \approx \mathbf{c}$. Since the event knowledge graph here has three types of relation,

$\mathbf{r}$ will belong to $\{\mathbf{r}_{temporal}, \mathbf{r}_{subevent}, \mathbf{r}_{causal}\}$. I also train a BiLSTM encoder to build event phrase embeddings **e1** and **e2**.

Intuitively, event nodes that are similar in semantics will have similar neighbor events in acquired knowledge graph, for example, both event *hurricane* and event *earthquake* are natural disasters, so they are likely to have similar "causality" events such as *people become homeless* and *houses collapse* as well as similar "subevents" such as *people seek shelter* and *rescue*. To see how learned event knowledge embeddings are different from traditional word embeddings, I apply Principal Component Analysis (PCA) and extract the first two components to visualize event nodes in a two-dimensional space. Visualization of sampled event words using traditional **word2vec** [Mikolov et al., 2013c] embeddings or using event knowledge graph embeddings is shown in Figure 6.2 and Figure 6.3 respectively. Overall, we can see both methods can group event words that are similar in semantics to clusters. In both figures, *disaster*, *earthquake* as well as *storm* are close to each other and *birthday*, *wedding* as well as *ceremony* are close to each other. Interestingly, *recession*, *bankruptcy* and *reform* are close in event knowledge graph embeddings but are relatively far to each other in word embeddings. One explanation is that these event words may be used differently in local context words but their neighbor events connected by temporal, subevent or causal relation edges are relatively similar.

6.3 Applying Event Embeddings to Relation Identification Tasks

To investigate whether the comprehensive event knowledge graph that includes all temporal, subevent and causal relations can further improve event-event relation identification tasks, I adopt **event relational links** and **event knowledge graph embedding** (Chapter 5) together with **averaged neighbor event embedding** as external event knowledge to event relation identification tasks used before. Specifically, for **event relational links**, for a pair of events to classify in benchmark datasets, I check if they match with event relation tuples of each relation type and encode the log values of counts in two directions in a vector. For **event knowledge graph embedding**, I use the trained BiLSTM event phrase encoder in the last section to get the event representation vector.

Method	RED	HiEve
Train and test on intra-sentence event pairs		
Basic BERT Classifier	61.8/52.3/56.6	46.7/52.4/49.4
+ Multi-relation links	69.0/54.5/60.9	50.3/50.6/50.5
+ Multi-relation Averaged Neighbor Emb.	65.0/53.6/58.7	48.5/54.5/51.3
+ Multi-relation KG Emb.	65.4/56.2/60.5	50.4/51.4/50.9
Train and test on cross-sentence event pairs		
Basic BERT Classifier	65.0/64.8/64.9	32.6/34.7/33.6
+ Multi-relation links	68.2/66.0/67.1	32.7/34.8/33.7
+ Multi-relation Averaged Neighbor Emb.	71.2/60.2/65.3	31.5/34.4/32.9
+ Multi-relation KG Emb.	68.8/63.1/65.8	33.6/32.0/32.8

Table 6.2: Subevent Relation Identification. P/R/F1 (%). Micro-average performance Parent-Child and Child-Parent on subevent relations.

Averaged neighbor event embedding: Neighbor events connected by different edges provide different aspects of the central event. For instance, neighbor events connected by temporal relation edges are regular consecutive events of the central event, which predicts possible future events that are likely to happen; neighbor events connected by subevent relation edges are children events which elaborate and expand the parent event; neighbor events connected by causal relation edges provide possible effects which specify consequences of the central event. Therefore, all neighbor events connected by different relation edges can provide meaningful content to enhance the representation of the central event. To encode neighborhood information differently, each event node has three channels where each channel encodes neighbors connected by one relation type (i.e., temporal, subevent or causal relation type). To do so, I retrieve all neighbor events connected by a relation type for a central event and calculate the weighted mean of word embeddings considering all the words in neighbor events[1]. The final event embedding representation is the concatenation of three channels.

Consistently, I concatenate two event word representations obtained by BERT encoder with either event knowledge embeddings using **knowledge graph embedding** or **averaged neighbor event embedding**. Results on subevent relation identification, temporal relation identification and

[1] `https://code.google.com/archive/p/word2vec/`

Method	RED	TimeBank
Train and test on intra-sentence event pairs		
Basic BERT Classifier	59.9/68.2/63.8	63.4/68.0/65.6
+ Multi-relation links	61.1/69.7/65.1	65.8/67.4/66.6
+ Multi-relation Averaged Neighbor Emb.	61.4/68.8/64.9	63.9/67.9/65.9
+ Multi-relation KG Emb.	59.3/71.7/64.9	62.6/70.8/66.4
Train and test on cross-sentence event pairs		
Basic BERT Classifier	38.4/37.4/37.9	53.3/42.1/47.1
+ Multi-relation links	52.4/42.3/46.8	52.6/48.6/50.5
+ Multi-relation Averaged Neighbor Emb.	41.0/42.3/41.6	56.7/44.5/49.9
+ Multi-relation KG Emb.	52.1/41.2/46.0	53.1/48.6/50.7

Table 6.3: Temporal Relation Identification. P/R/F1 (%). Micro-average performance on Before and After temporal relations.

Method	RED	ESC
Train and test on intra-sentence event pairs		
Basic BERT Classifier	63.9/66.6/65.2	44.9/52.2/48.3
+ Multi-relation links	63.6/66.8/65.2	45.3/54.4/49.5
+ Multi-relation Averaged Neighbor Emb.	65.3/67.2/66.2	46.6/53.1/49.7
+ Multi-relation KG Emb.	64.0/64.4/64.2	44.2/53.2/48.3
Train and test on cross-sentence event pairs		
Basic BERT Classifier	22.5/15.1/18.1	21.6/28.5/24.6
+ Multi-relation links	28.5/29.4/28.9	21.1/35.8/26.6
+ Multi-relation Averaged Neighbor Emb.	22.8/19.3/20.9	22.5/29.8/25.6
+ Multi-relation KG Emb.	27.3/29.4/28.3	23.3/28.0/25.4

Table 6.4: Causal Relation Identification. P/R/F1 (%). Micro-average performance on Cause-Effect and Effect-Cause relations.

causal relation identification are shown in Table 6.2, Table 6.3 and Table 6.4. We can see that incorporating learned event knowledge in either way improves the baseline BERT classifier.

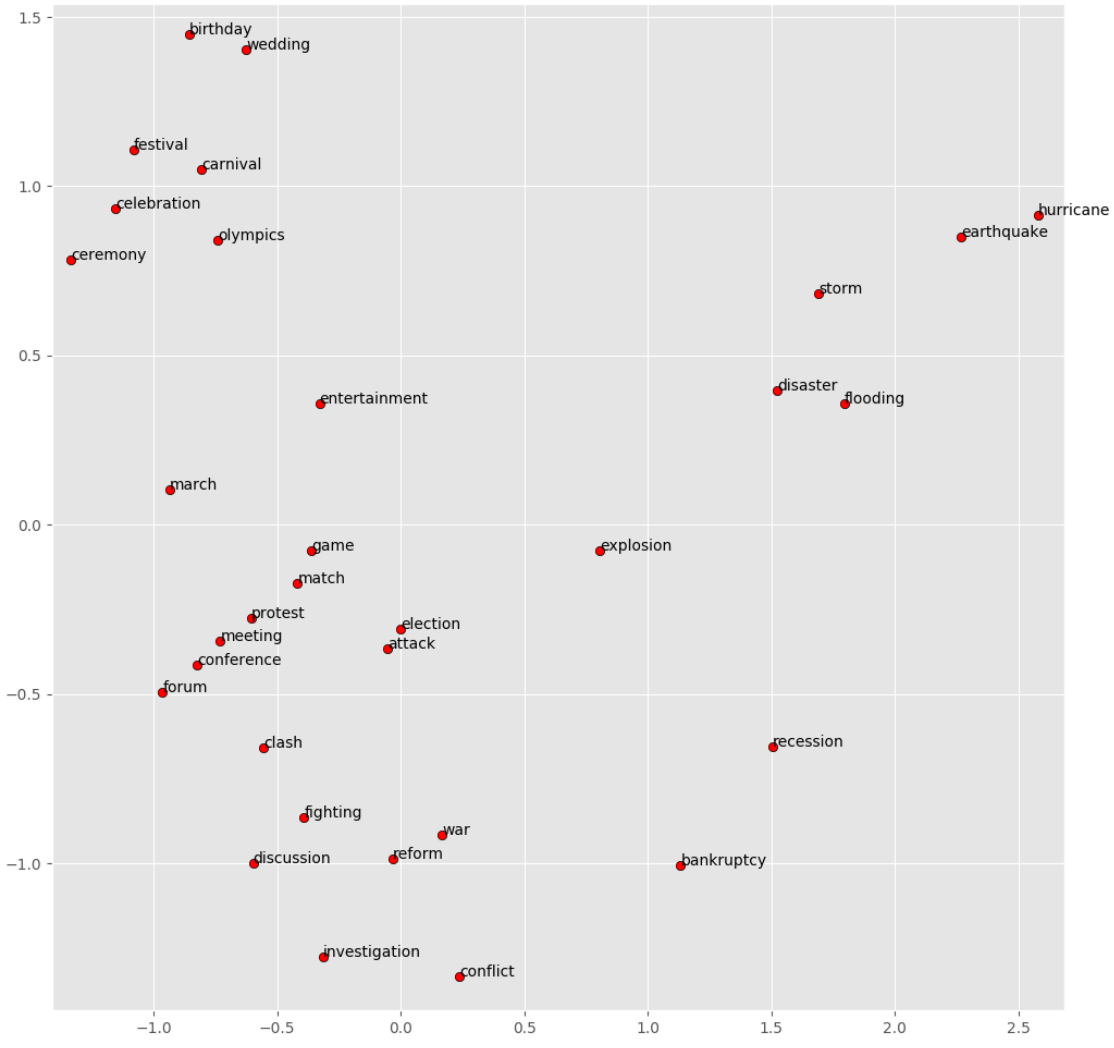

Figure 6.2: Visualization of Event Words Using Word Embeddings [Mikolov et al., 2013c]

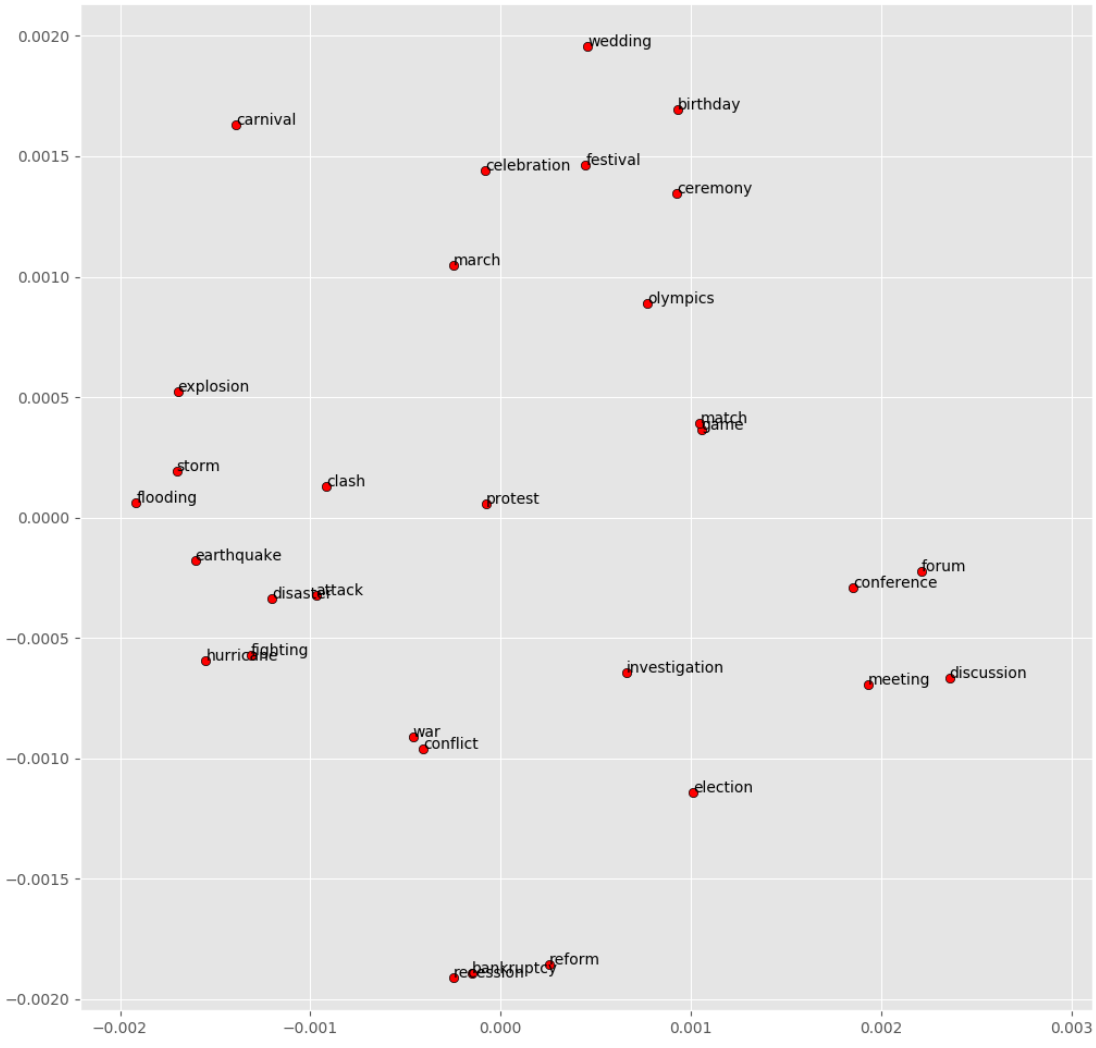

Figure 6.3: Visualization of Event Words Using Knowledge Graph Embedding

7. WEAKLY-SUPERVISED FINE-GRAINED EVENT RECOGNITION ON SOCIAL MEDIA TEXTS FOR DISASTER MANAGEMENT[1]

Recognizing events of specified types (e.g., conflict events, life-threatening events) is a fundamental problem in event tracking and risk management. It is challenging because one event might have various expressions and belong to different event types depending on different contexts. Taking the event word "dead" for example, in addition to the meaning of "losing life" that belongs to Casualty type, "dead" is also frequently used to refer to phones being out of power. Recognizing events in specified types automatically and accurately is an important topic in information extraction. Especially during natural disasters, to meet the high communication and information sharing demands during disasters, the local authorities, responders, and victims frequently use social media platform to provide real-time situation updates. How to recognize fine-grained types of life-threatening emergency events in a timely manner is critical for reducing life and property losses.

Therefore, I propose a weakly supervised approach for rapidly building high-quality classifiers that label each individual Twitter message with fine-grained event categories. Figure 7.1 gives an overview of my weakly-supervised learning approach with three phases. In phase one, I quickly create high-quality labeled data. Specifically, I conduct clustering-assisted manual word sense disambiguation on event keyword identified noisy tweets, to significantly clean and improve the quality of automatically labeled tweets. In phase two, I train a multi-channel BiLSTM classifier using tweets together with their context tweets and reply tweets. In phase three, I iteratively retrain the multi-channel classifier to further improve its event recognition performance.

[1]Reprinted with permission from "Weakly-supervised Fine-grained Event Recognition on Social Media Texts for Disaster Management" by Wenlin Yao, Cheng Zhang, Shiva Saravanan, Ruihong Huang and Ali Mostafavi, 2020. Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 01, pp. 532-539).

7.1 Event Categories and Event Keywords

Disaster management generally consists of four phases - mitigation, preparedness, response, and recovery. I focus on identifying events during the *response* phase of disasters, which is arguably the most crucial and time-critical part of emergency management. Based on an existing event ontology for hurricanes [Huang and Xiao, 2015], I identified nine types of events, including three types of human activity events and six types of built environment related events, as briefly described below.

Human activities. 1) Preventative measure (PRE). People look for shelters or process evacuation; Any flood-proof processes (e.g., building waterproof facilities, etc.). 2) Help and rescue (RES). People provide, receive, or seek face to face help in disastrous environments, including indirect help such as donating money, supply, and providing services. 3) Casualty (CAS). Disaster-caused death, injury, hurt, etc.

Built environment. 4) Housing (HOU). Reporting emergencies of a house, apartment, home, etc. 5) Utilities and Supplies (UTI). Problems with heating, gas, water, power, communication facility, food, grocery stores, etc. 6) Transportation (TRA). The impact on the traffic, bus services, or the closure of a road, airport, highway, etc. 7) Flood control infrastructures (FCI). The impact on or damage to the reservoir, bayou, canal, dam, etc. 8) Business, Work, School (BWS). The changes of schedule, e.g., business closed/open, school closed/open, etc. 9) Built-environment hazards (HAZ). The damage or risks that may cause injury or death related to the built environment, such as fire, explosion, contamination, electric shock, debris, etc.

Meanwhile, the event ontology [Huang and Xiao, 2015] contains event keywords, and I selected at most five keywords for each event category that are not specific to any particular hurricane or location, e.g., keywords “evacuate” and “shelter” for the category of Preventive measure (PRE), and “help” and “rescue” for Help and rescue (RES), etc.

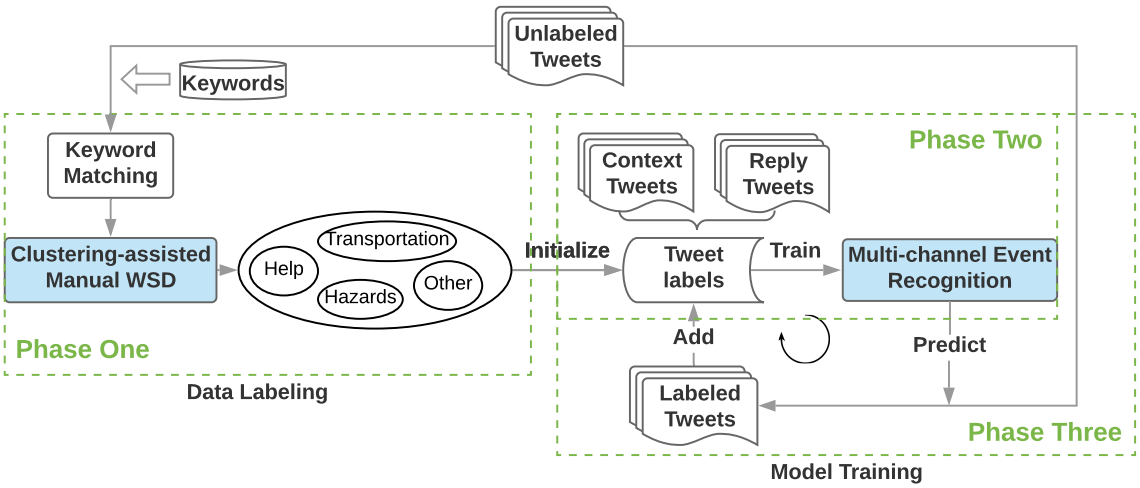

Figure 7.1: Overview of the Weakly-supervised Event Recognition System for Disaster Management. Reprinted with permission from Yao et al. [2020b].

7.1.1 Phase One: Rapid Data Labeling via Clustering Assisted Manual Word Sense Disambiguation

For each event category, I first retrieve tweets containing a predefined event keyword and then apply a clustering algorithm to form tweet clusters. To facilitate manual word sense disambiguation, I rank tweet clusters based on their sizes (number of tweets) and then ask a domain expert to judge whether a cluster (from largest to smallest) shows the pertinent meaning of an event keyword by inspecting five example tweets randomly sampled from the cluster. The annotator stops scrutiny once 20 pertinent clusters are identified for each event category[2]. After cleaning, around a third to half of keyword identified tweets were removed. Specifically, 6.6K out of 15.2K keyword identified tweets and 5.8K out of 17.5K keyword identified tweets were removed in the *Harvey* and *Florence* datasets respectively.

Next, I describe the clustering algorithm used here, the Speaker-Listener Label Propagation Algorithm (SLPA) [Xie et al., 2011].

[2]If a cluster adopts a word sense that is irrelevant to any event category, I assign the catch-all label "Other" to it and later use tweets in such clusters as negative training instances.

7.1.1.1 The Clustering Algorithm

The SLPA algorithm is initially introduced to discover overlapping communities in social user networks, where one user may belong to multiple communities. The basic idea of SLPA is to simulate people's behavior of spreading the most frequently discussed topics among neighbors. I choose SLPA for two reasons. First, SLPA is a self-adaptation model that can automatically converge to the optimal number of communities, so no pre-defined number of communities is needed. Second, a tweet during natural disasters may mention more than one event, which corresponds to one user belonging to multiple communities. SLPA has been shown one of the best algorithms for detecting overlapping communities [Xie et al., 2013].

Clustering with Graph Propagation: SLPA is essentially an iterative algorithm. It first initializes each node as a cluster by itself. In listener-speaker propagation iterations, each node will be chosen in turn to be either a listener or a speaker. Each time, a listener node accepts the label that is the most popular among its neighbors and accumulates such knowledge in the memory. And a speaker advocates one label based on the probability distribution updated in its memory. Finally, based on the memories, connected nodes sharing a label with the probability over a threshold are grouped together and form a community.

I modified the original SLPA to make it suitable for clustering Twitter messages. Formally, given a set of tweets, I construct an undirected graph $G(V, E)$, where V represents all tweets and E represents weighted edges between nodes. The weight of an edge e between two tweets u and v is calculated based on content similarity of the two tweets. In label propagation, I consider weighted voting to determine the cluster of a tweet.

The Similarity Measure: Determining similarities between nodes is important for clustering algorithms. However, Twitter messages are informal and often contain meaningless words, therefore, I aim to first select important words before calculating content similarities between tweets. Recently, [Conneau et al., 2017] proposed an approach for learning universal sentence representations using the Stanford Natural Language Inference (SNLI) dataset [Bowman et al., 2015] and demonstrated its effectiveness in reasoning about semantic relations between sentences. I notice

that Twitter messages and SNLI data have two common characteristics: short sentences in a casual language. Hence, I apply their learned sentence representation constructor to tweets for identifying important words.

Specifically, for a given tweet with T words $\{w_t\}_{t=1,2,...,T}$, I applied the pre-trained Bi-directional LSTMs [Conneau et al., 2017] to compute T hidden vectors, $\{h_t\}_{t=1,2,...,T}$, one per word. Next, for each dimension, I determine the maximum value over all the hidden vectors $\{h_t\}_{t=1,2,...,T}$. The importance score for a word w_t is calculated as the number of dimensions where its hidden vector h_t has the maximum value divided by the total number of dimensions. Then, I select words having importance scores $\geq$ the average importance score (1.0 / the number of words) as important words. For example, in the following tweet, the bolded words are selected: *It has **started** a **fundraiser** for **hurricane** Harvey **recovery** efforts in **Houston**, you can **donate** here.*

I calculate the similarity score between two tweets by considering only selected words shared by two tweets. Empirically, I found this similarity measure performs better than the straightforward cosine similarity measure considering all words. Specifically, the similarity score between two tweets u and v is the number of common words / (length of $u \times$ length of v). To construct the tweet graph, I create an edge between two tweets when they share two or more selected words and the edge weight is their similarity score.

7.1.2 Phase Two: Multi-channel Tweet Classification

The most unique characteristic of social media is the network structure which not only connects users (e.g., friend network or follower network), but also makes Twitter messages connected. Therefore, I exploit other related tweets for enhancing the representation of a target tweet. In particular, I found the immediately preceding context tweets and reply tweets useful.

First, the past tweets written by the same user provide additional evidence for an event recognition system to infer the event topic of the current tweet. Interestingly, I observe that the event topic engaging a user's attention is usually consistent within a small time window, as shown in the upper example of Figure 1.6 where the two relevant context tweets are within 2 minutes. I further observe that the topic relatedness between the target tweet and context tweets decreases quickly over time.

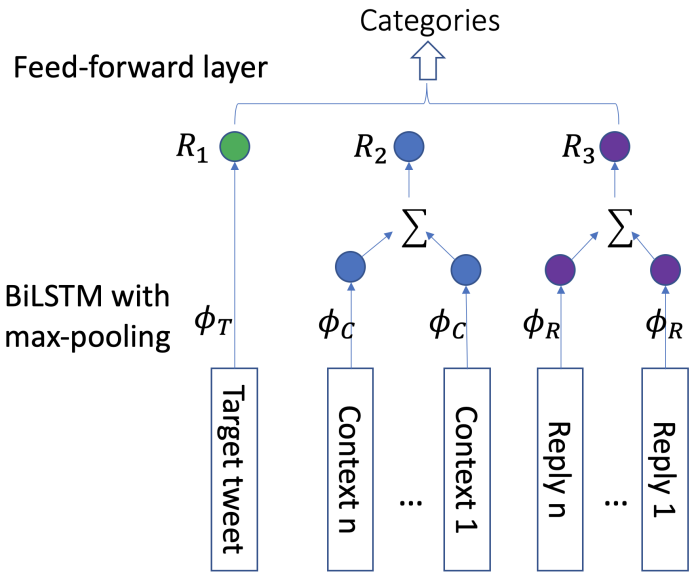

Figure 7.2: BiLSTM Classifier using Context and Reply Enriched Representation. Reprinted with permission from Yao et al. [2020b].

In my experiments, I only consider a relatively small number of context tweets, specifically five of the preceding tweets. In addition, I assign a weight to a context tweet as $w_i = 0.8^{m_i}$, where m_i is the time distance (in minutes) between the i^{th} context tweet and the target tweet.

Second, reply tweets usually provide information that is hidden in the original tweet, as shown in the lower example of Figure 1.6. But compared to regular Twitter posts, replies are much noisier. To select the most informative reply tweets for a given target tweet, I rank replies according to the number of common words they share with the target tweet and pick a small number of them from the top, specifically at most five replies.

Figure 7.2 shows the overall structure of the classifier.

$$
\begin{aligned}
R_1 &= \Phi_T(tw^{target}) \\
R_2 &= \frac{1}{\sum w_i} \sum_{i=1}^{N} w_i \cdot \Phi_C(tw_i^{context}) \\
R_3 &= \frac{1}{M} \sum_{i=1}^{M} \Phi_R(tw_i^{reply}) \\
R_{all} &= [R_1, R_2, R_3]
\end{aligned}
\tag{7.1}
$$

Specifically, I apply three separate BiLSTM encoders [Graves and Schmidhuber, 2005] with max-pooling [Collobert and Weston, 2008] to obtain sentence embeddings for the target tweet, context tweets and reply tweets (i.e., Φ_T, Φ_C, Φ_R). Then, the final enriched representation of the target tweet (R_{all}) is the concatenation of the target tweet embedding (R_1), weighted average of context tweet embeddings (R_2), and unweighted average of reply tweet embeddings (R_3).

On top of R_{all}, I apply a feedforward neural net to directly map R_{all} to 10 classes (9 event categories + Other). I optimizes a multi-label one-versus-all loss based on max-entropy, considering that one tweet may belong to multiple event categories. To deal with imbalanced distributions of event categories, I re-scale the prediction loss of each class (proportional to $\frac{1}{ClassSize}$) so that smaller classes are weighted more heavily in the final loss function. For all BiLSTM encoders, I use one hidden-layer of 300 units, pre-trained GloVe [Pennington et al., 2014] word embeddings of 300 dimensions, Adam optimizer [Kingma and Ba, 2015] with a learning rate of 0.0001.

In training, to compete with positive training instances (tweets labeled with any event category), I randomly sample unlabeled tweets equal to the sum of labeled tweets in size and use them as negative training instances (the category Other), to reflect the fact that there are generally more tweets reporting no event.

7.1.3 Phase Three: Improve Coverage with Bootstrapping Learning

After the first two phases, we have labeled tweets by conducting time-efficient clustering-assisted WSD on event keyword identified tweets and have used these quickly labeled tweets to

Category	PRE	RES	CAS	HOU	UTI	TRA	FCI	BWS	HAZ	Other	Sum
Harvey (Aug.28 1:00-2:00 pm)											
Amount	374	1092	43	142	270	225	73	501	30	9165	11782
Percentage	3.2%	9.3%	0.4%	1.2%	2.3%	1.9%	0.6%	4.3%	0.3%	77.8%	100%
Florence (Sept.17 1:00-1:30 pm)											
Amount	69	490	120	28	146	85	8	80	23	3031	4059
Percentage	1.7%	12.1%	3%	0.7%	3.6%	2.1%	0.2%	2%	0.6%	74.7%	100%

Table 7.1: Annotation: Number of Tweets in Each Event Category. Reprinted with permission from Yao et al. [2020b].

train the multi-channel event recognizer. However, all the labeled tweets yielded in phase 1 contain a predefined event keyword, while many event categories may have cases that do not contain a keyword. Therefore, I further exploit bootstrapping learning and iteratively improve the coverage of the multi-channel classifier.

Specifically, I apply the initial multi-channel classifier on unlabeled tweets and label new tweets for each event category. Newly labeled tweets together with their context tweets and replies are used to retrain the model. To enforce the classifier to look at new content words other than event keywords, I randomly cover 20% of keywords occurrences in every training epoch, inspired by [Srivastava et al., 2014]. In order to combat semantic drifts [McIntosh and Curran, 2009] in bootstrapping learning, I initially apply a high confidence score for selecting newly labeled tweets used to retrain the classifier and lower the confidence score gradually. Specifically, the confidence score was initially set at 0.9 and lowered by 0.1 each time when the number of selected tweets is less than 100. The bootstrapping process stops when the confidence score decreases to 0.5[3].

7.2 Experiments and Results

7.2.1 Data Sets

I apply the approach to datasets for two hurricanes, *Harvey* (the primary dataset) and *Florence* (the second dataset). Hurricane *Harvey* struck the Houston metropolitan area and Southeast Texas

[3]In my experiment, the bootstrapping process stopped after 9 iterations, and each iteration took around 10 minutes using a NVIDIA's GeForce GTX 1080 GPU.

Row	Method	PRE	RES	CAS	HOU	UTI	TRA	FCI	BWS	HAZ	Macro Average
1	Keyword Matching	73.9	56.6	26.2	36.4	54.3	38.0	54.4	55.5	43.1	51.1/52.5/51.8
2	LDA	39.4	41.3	4.9	8.5	19.8	28.8	40.4	17.7	25.5	19.6/42.6/26.8
3	Guided LDA	43.4	45.8	10.1	8.6	21.1	40.7	53.4	20.4	24.5	25.1/45.2/32.3
4	SLPA	61.4	61.3	18.9	23.1	36.4	36.2	56.5	44.4	23.3	39.6/48.1/43.4
Seed with Keyword Identified Tweets with no Cleaning											
5	Basic Classifier	82.6	63.8	18.8	36.9	60.2	36.8	61.7	61.0	45.5	50.3/60.5/54.9
6	+ bootstrapping	82.6	64.1	20.1	37.3	60.6	36.5	62.8	60.6	45.7	50.2/61.2/55.3
7	Multi-channel Classifier	**84.3**	68.6	22.1	37.1	60.5	40.5	62.3	62.1	45.7	50.5/64.1/56.5
8	+ bootstrapping	84.1	69.1	22.6	36.4	59.6	42.1	63.1	60.4	46.4	49.4/**65.8**/56.4
Seed with Keyword Identified Tweets Cleaned by Clustering-assisted WSD											
9	Basic Classifier	82.6	68.4	34.4	45.1	65.8	56.4	63.3	68.4	49.0	68.3/57.2/62.3
10	+ bootstrapping	83.5	68.8	36.9	45.0	65.8	58.4	66.7	68.5	51.9	67.1/59.9/63.3
11	Multi-channel Classifier	82.8	68.0	36.7	**47.6**	65.1	57.2	63.3	67.8	56.5	72.5/57.0/63.8
12	+ bootstrapping	83.9	67.8	36.7	45.7	**66.1**	61.3	**74.8**	69.1	**57.7**	70.1/61.6/**65.5**
13	Supervised Classifier	80.8	**72.2**	**48.0**	45.3	56.3	**67.9**	65.9	**71.2**	45.3	**73.2**/53.6/61.9

Table 7.2: Experimental Results on Hurricane Harvey: F1-score for each event category and macro-average Precision/Recall/F1-score (%) over all categories. Reprinted with permission from Yao et al. [2020b].

in 2017, and ranks as the second costliest hurricane ($125 billion in damage) on record for the United States [National Hurricane Center, 2017]. Hurricane Florence also caused severe damage (more than $24 billion) in the North and South Carolina in 2018. To retrieve tweets in affected areas, I consider two constraints in twitter crawling using GNIP API [Twitter, 2019]: 1) a tweet has the geo-location within affected areas (Houston or major cities in Carolinas) or 2) the author of a tweet has his/her profile located in affected areas. Since I aim to recognize original tweet messages reporting events for disaster management purposes, I only consider original tweets as target tweets for classifications across all the experiments and I ignore retweets and reply tweets.

To create the official evaluation data (details in the next section), we exhaustively annotated all the tweets posted from 1:00 to 2:00 pm, August 28, 2017 for *Harvey* and from 1:00 to 1:30 pm, September 17, 2018 for *Florence*, both among the most impacted time periods for the two hurricanes. For training both my systems and the baseline systems, I used around 65k and 69.8k unlabeled tweets for *Harvey* and *Florence* respectively that were posted 12 hours (half a day) preceding the test time period and are therefore strictly separated from the tweets used for evaluation.

7.2.2 Human Annotations for Evaluation

In order to obtain high-quality evaluation data, I trained two annotators and refined annotation guidelines for several rounds. A tweet is annotated with an event category if it directly discusses events of the defined category, including sharing information and expressing opinions. A tweet may receive multiple labels if it discusses more than one event and the events are of different types. If one tweet does not discuss any event of an interested type, I label it as *Other*.

I first asked the two annotators to annotate a common set of 600 tweets from the *Harvey* set and they achieved a substantial kappa score of 0.67 [Cohen, 1968]. I then split the remaining annotations evenly between the two annotators. The distributions of annotated tweets are shown in Table 7.1.[4] Consistent across the two considered hurricane disasters, tweets describing interested events cover only around one quarter of all posted tweets and their distributions over the event categories are highly imbalanced.

7.2.3 Unsupervised Baseline Systems

Keyword matching: labels a tweet with an event category if the tweet contains any keyword in the event category. A tweet may be assigned to multiple event categories if the tweet contains keywords from more than one event category.

Topic modeling Approaches: Probabilistic topic modeling approaches have been commonly used to identify latent topics from a collection of documents. I assign each topic to an event category if the top ten words of a topic ranked by word probabilities contain any keyword of the category. A topic may be assigned to multiple event categories if its top ten words contain keywords from more than one category. Given a new tweet, I infer its topics and assign the event labels of the most significant topic. I implement two topic modeling approaches. **LDA (Latent Dirichlet Allocation)** [Blei et al., 2003] assumes a document can be represented as a mixture over latent topics, where each topic is a probabilistic distribution over words. **Guided LDA** [Jagarlamudi et al., 2012] is a stronger version of LDA, that incorporates my predefined event keywords to guide the topic

[4] A small number of tweets were annotated with more than one event category, 290 (11%) and 57 (6%) tweets for *Harvey* and *Florence* datasets respectively.

discovery process. For fair comparisons, I also apply important words selection used in my system for LDA and GuidedLDA[5]. Note that both approaches require pre-defining the number of topics, which is hard to estimate, I set this hyper-parameter as 100 in my experiments.

SLPA: I also apply the adapted SLPA clustering algorithm to form clusters and assign each cluster to an event category if the top ten words in a cluster ranked by word frequencies contain any keyword of the category. Given a new tweet, I identify its neighbor tweets using the same similarity measure I used for clustering in phase one and label the tweet with the majority event label over its neighbors.

7.2.4 Results on Hurricane *Harvey*

Table 7.2 shows the experimental results. The first section shows performance of baseline systems. Among the four baselines, the simple keyword matching approach (row 1) performs the best, and the clustering algorithm SLPA (row 4) outperforms both LDA-based approaches (row 2 & 3). The event recognition performance of these mostly unsupervised systems is consistently low, presumably due to their incapability to resolve severe lexical ambiguities in tweets.

The second section of Table 7.2 shows results of four classifiers that directly use keyword identified noisy tweets with no cleaning for training. Row 5 shows the results of the basic classifier considering the target tweet only. Row 7 shows the results of the multi-channel classifier that further considers contexts and replies, which yields a small recall gain compared to row 5. Row 6 & 8 show the results of the two classifiers after applying bootstrapping learning, which further improves the recall a bit. However, the precision of all the four classifiers is around 50% similar to the keyword matching baseline and consistently unsatisfactory.

The third section of Table 7.2 shows results of the same set of classifiers but using clustering-assisted WSD cleaned tweets for training. Compared to its counterpart trained using noisy tweets (row 5), the precision of the basic classifier (row 9) improves significantly by 18%. With a small drop on recall, the overall F-score improves by 7.4%. The multi-channel classifier (row 11) further improves the precision with an almost identical recall. Bootstrapping learning improves the recall

[5]I also tried LDA and Guided LDA without important words selection which yields a much worse performance.

of both classifiers. The full system (row 12) outperforms its counterpart trained using noisy tweets (row 8) by over 20% in precision and 9% in F-score. Meanwhile, using a little supervision, the rapidly trained weakly supervised system greatly outperforms the unsupervised baseline systems, yielding 20% (or more) and 15% (or more) of increases in precision and F-score respectively.

Comparisons with Supervised Learning: I train and evaluate a supervised classifier (multi-channel) using annotated tweets under the 10-fold cross validation setting. The results of the supervised classifier are shown in the last row of Table 7.2. Compared to the supervised classifier, the weakly supervised approach yields a recall gain of 8% with a slightly lower precision, and improves the overall F-score by 3.6%. Note that around 50 person-hours were needed to annotate over 11K tweets following the normal tweet-by-tweet annotation process, while my data labeling method only required 1-2 person-hours for clustering-assisted WSD. Considering that a large number of tweets are time-consuming to annotate, I conducted another group of experiments that gradually add annotations in training to see how the size of training data affects the performance. Specifically, under 10-fold cross validation, I randomly sample a certain percentage of tweets from nine training folds as training data, ranging from 0.1 to 0.9 in increments of 0.1. The learning curve (Figure 7.3) is steep in the beginning and then levels out as the remaining 70% of annotated tweets (around 7K tweets) were continuously appended, which shows that the normal annotation method may create many redundant annotations.

7.2.5 Results on Hurricane *Florence*

Table 7.3 shows the results. Similar to Hurricane *Harvey*, clustering-assisted WSD clearly improves the precision of the trained classifier for Hurricane *Florence* as well. Enriching tweet representations and conducting bootstrapping learning further improve the performance of the full system, which clearly outperforms the supervised classifier.

7.2.6 Analysis

For Hurricane *Harvey*, I applied the full system to label tweets posted right after the test hour. Figure 7.4 plots the number of tweets detected for each hour. Overall, the clear low point corre-

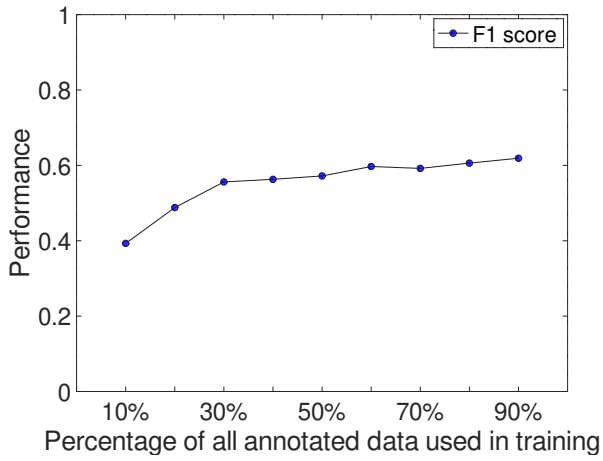

Figure 7.3: Learning curve of 10-fold cross validation. Reprinted with permission from Yao et al. [2020b].

sponds with the day-night shift. Taking a closer look at the curve for the flood control infrastructure category, we can see an obvious burst at 8 pm Aug.28, 2017, triggered by an official update on water release of two major reservoirs, as well as a burst at 10 am Aug.29, triggered by the collapse of a bridge over Greens Bayou, with example tweets shown in Figure 7.5.

7.3 Conclusion

I have presented a weakly supervised event recognition system that can effectively recognize fine-grained event categories for individual tweet messages. I highlight the novel clustering-assisted manual word sense disambiguation data labeling method that is time-efficient and significantly improves the quality of event keyword identified texts. The evaluation on two hurricanes show the effectiveness and robustness of the overall approach. The weakly supervised system can be easily adapted to other disaster types (e.g., earthquake, tsunami, etc.) with a relevant event ontology to support real-time disaster management.

	Macro Average
Keywords	43.7/46.9/45.3
with no Cleaning	
Basic Classifier	40.8/47.9/44.1
+ bootstrapping	39.8/52.8/45.4
Multi-channel Classifier	43.1/48.7/45.8
+ bootstrapping	41.2/52.6/46.2
with Clustering-assisted WSD	
Basic Classifier	67.8/49.6/57.3
+ bootstrapping	63.4/54.9/58.8
Multi-channel Classifier	**70.3**/50.2/58.5
+ bootstrapping	65.1/**55.1**/**59.7**
Supervised Classifier	57.8/40.9/47.9

Table 7.3: Experimental Results on Hurricane Florence (Precision/Recall/F1-score %). Reprinted with permission from Yao et al. [2020b].

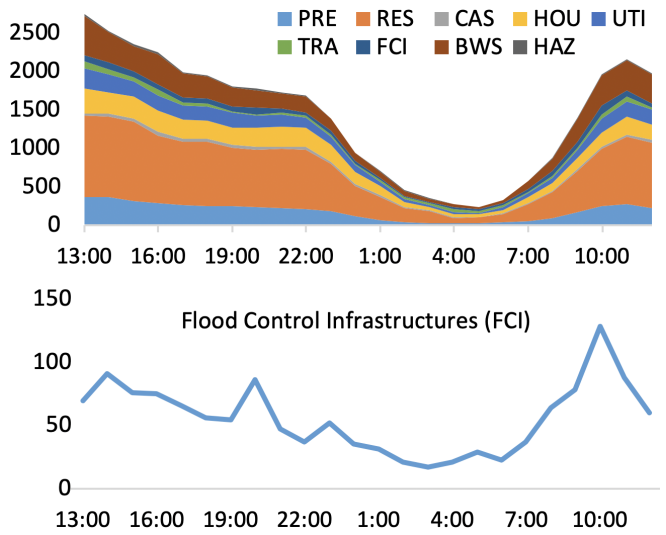

Figure 7.4: Curves for all the categories (Upper) and for Flood Control Infrastructures only (Lower). Reprinted with permission from Yao et al. [2020b].

HAPPENING NOW: @hcfcd live update on Addicks Reservoir and certain levees. Watch now on TV or here. One of the dams they want to discharge is near me.	BREAKING: The levee at Columbia Lakes has been breached! GET OUT NOW! PLEASE BE SAFE! A bridge has collapsed at Greens Bayou. Be careful!

Figure 7.5: Example tweets sampled from two bursts. Reprinted with permission from Yao et al. [2020b].

8. CONCLUSION

In this chapter, I will first summarize the contributions of four proposed weakly-supervised learning systems including three systems that extract rich and diverse event relational knowledge and one system that recognizes life-threatening events on social media during natural disasters. Finally, I will discuss possible future research directions.

8.1 Research Summary

I began this dissertation with the significance of interpreting events and event-event relations in computational applications to understand the dynamics of the world. I demonstrated the fundamental need for event relational knowledge to facilitate comprehensive reasoning in unstructured texts and time-critical needs to achieve automatic recognition of life-threatening events based on social media messages during natural disasters.

Traditional supervised learning approaches recognize text-internal event relations heavily relying on human-annotated data. Labeled data are often limited in size and context diversity, as labeling process requires tremendous efforts of well-trained human annotators. Therefore, classifiers trained in a supervised way has difficulty to cover large and diverse event contexts and cannot scale well on big data to acquire event knowledge.

To address the limitations of supervised-learning systems on knowledge acquisition, I proposed three weakly-supervised approaches to acquire event temporal and subevent knowledge. For event temporal knowledge acquisition, I developed two approaches, where the first approach bootstraps event temporal pairs using a CNN intra-sentence classifier while the second approach distills even temporal chains from narratives that are extracted using a document level narrative recognizer. Specifically, the first weakly supervised bootstrapping approach focuses on knowledge acquisition from intra-sentences. It learns both regular event pairs and a contextual temporal relation classifier, by exploring the observation that regular event pairs tend to show a consistent temporal relation despite of their diverse contexts. The evaluation shows that the learned regular event pairs are of

high quality and rich in commonsense knowledge and domain knowledge. The second approach focuses on knowledge acquisition from cross-sentences using discourse structure. It leverages the double temporality characteristic of narrative texts and acquires event temporal knowledge across sentences in narrative paragraphs. My method acquires event temporal knowledge from narratives that are automatically identified using narratology principles, which substantially reduces the reliance on human-annotated data. The event temporal knowledge distilled from narrative texts has a high quality, which was shown useful to improve temporal relation classification and outperform several neural language models on the narrative cloze task.

Subevent relations that indicate that several subevents happen as parts of the parentevent spatio-temporally also widely exist among events. For acquiring event subevent knowledge, I proposed another weakly supervised learning framework that starts by collecting rich and high-quality weak supervision from definition-guided semantic check. Specifically, I first obtained 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, I collected 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. In this way, I built the first large scale subevent knowledge base containing 239K subevent tuples. The evaluation shows that our acquired subevent knowledge has an overall accuracy of 90.1% and has great coverage of different event types. Empirical experiments on three benchmark datasets show our learned knowledge is beneficial to subevent relation recognition and discourse parsing.

After acquiring all event knowledge, I constructed a general event knowledge graph that has 126K event nodes, 174K *happens-before* edges, 239K *parent-subevent* edges and 49K *cause-effect* edges. I further explored several learning methods to project each event phrase node to a distributed vector representation so that the knowledge graph structure is encoded. The visualization and evaluation provide us insights of acquired distributed event representations.

Event detection is also important in world modeling but current event detection systems also

mainly rely on supervised-learning approaches. However, under some application circumstances, such as event detection during emergent natural disasters, it is almost impossible to spend days or weeks to collect labeled data to train a supervised system. Thus, I presented a weakly supervised event recognition system that can effectively recognize fine-grained event categories for individual tweet messages. The novel approach - clustering-assisted manual word sense disambiguation data labeling method - is time-efficient and significantly improves the quality of event keyword identified texts. The evaluation on two hurricanes show the effectiveness and robustness of the overall approach. Because of its weakly-supervised flavor, my system can be easily adjusted to a new natural disaster type (e.g., earthquake, tsunami, etc.) by changing to a new event ontology, which supports real-time disaster response and management based on social media during devastating disasters.

8.2 Looking Forward

Experiments of my research have shown that event relational knowledge can help to improve a range of NLP tasks. However, how to acquire and use event knowledge more precisely and effectively is still a challenging and open-ended question. In the following sections, I will discuss some specific research directions and present potential solutions that may lead to better event knowledge acquisition and usage.

8.2.1 Improving Accuracy and Coverage of Event Knowledge

The first important research question is how to further improve the quality of acquired event knowledge. The key task of acquiring event knowledge is to first recognize semantic relations among events from various contexts and then to distill event knowledge to more general conceptual statements. My proposed weakly-supervised approaches rely on *distant supervision* to automatically obtain relation instances from big corpus so that we can train a classifier to recognize particular event relations. However, training instances obtained using *distant supervision* inevitably contain noise. Therefore, we can apply some noise reduction methods [Roth et al., 2013](e.g., at-least-one models [Riedel et al., 2010, Yao et al., 2010], topic-based models [Alfon-

seca et al., 2012] or pattern correlations models [Takamatsu et al., 2012]) to improve the quality of automatically generated data, which will consequently improve the performance of the classifier.

The second important topic is how to improve the coverage of acquired event knowledge. Even I have constructed a knowledge base containing hundreds of K event knowledge tuples, it is still very sparse in real-world applications. For example, many observed event phrases in real-world applications may not appear in my acquired event ontology knowledge. One solution is that we can consider more text genres and larger text corpus in knowledge acquisition and even use multilingual data resources to acquire knowledge. Another solution is that we can apply a better named entity recognizer that considers more named entity types to generalize event arguments. Or we can build an event knowledge retrieval system to retrieve paraphrases or synonyms of the non-observed event representations from acquired event knowledge. For example, with a knowledge retrieval system, we can retrieve *building collapse* from our acquired event knowledge for a non-observed event phrase *mansion collapse* and know *flooding* may cause *mansion collapse*. To do so, we can either train a classifier based on annotated event paraphrase pairs or use semantic similarity measures (e.g., use pretrained word embeddings) to retrieve similar event phrases from acquired knowledge.

8.2.2 Learning Better Representations on Event Knowledge Graph

The second critical question is that how to use acquired event knowledge more effectively. After acquiring event knowledge graph, I mainly consider static approaches (i.e., relation links count and knowledge graph embeddings) that first encode event graph knowledge into knowledge vectors and then incorporate such knowledge vectors to event context-specific representations to improve event relation identification performance. However, sentence occurrences of the same event might emphasize different aspects of the event. Take the event *hurricane* as an example, we want the neural network to focus more on "happens-after" and "cause-effect" event neighbors when its sentence talks about the consequences of *hurricane*, but focus more on "subevent" neighbors when its sentence is about how people react during *hurricane*. Recently, Graph Neural Networks (GNNs) have been successfully applied to tackle problems where data are represented

in the form of graphs. GNNs consist of an iterative aggregation process to encode the node states dynamically, which effectively propagates information from neighbors. Therefore, inspired by recent advances of context-dependent word representation learning, one way to learn better event representations is to learn context-dependent event representation using GNNs. By conditioning on context words in different sentences, one graph neural network can dynamically assign weights to aggregate information from the neighbor events. This approach can also help with event word sense disambiguation.

8.2.3 Applying Event Knowledge to Other NLP Applications

In this dissertation, I have evaluated on applications directly relevant to events (i.e., narrative cloze tasks, event relation identification tasks, and discourse relation classification tasks) to show the usefulness of acquired event knowledge. As we know, events broadly exist in all forms of natural language usage, so event relational knowledge may benefit other high-level NLP applications (e.g., question answering, dialogue systems, machine reading comprehension, etc.) that are not directly relevant to events. For instance, if a human user tells the dialogue system that he/she recently suffers from insomnia and the dialogue system knows "take medicine normally *happens_after* insomnia", it probably can suggest the user see a doctor and take some medicine. Therefore, how to incorporate or apply event knowledge to many other NLP tasks is an open-ended question to investigate.

REFERENCES

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

Enrique Alfonseca, Katja Filippova, Jean-Yves Delort, and Guillermo Garrido. Pattern learning for relation extraction with hierarchical topic models. 2012.

Foteini Alvanaki, Michel Sebastian, Krithi Ramamritham, and Gerhard Weikum. Enblogue: emergent topic detection in web 2.0 streams. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 1271–1274. ACM, 2011.

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

Allison Badgett and Ruihong Huang. Extracting subevents via an effective two-phase approach. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 906–911, 2016.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet Project. In *In Proceedings of COLING/ACL*, pages 86–90, 1998.

Mieke Bal. *Narratology: Introduction to the theory of narrative*. University of Toronto Press, 2009.

Brandon Beamer and Roxana Girju. Using a bigram event model to predict causal potential. In *CICLing*, pages 430–441. Springer, 2009.

Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on twitter. *Icwsm*, 11(2011):438–441, 2011.

Adam L Berger, Vincent J Della Pietra, and Stephen A Della Pietra. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71, 1996.

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

Steven Bethard and James H Martin. 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, 2008.

Ann Bies, Zhiyi Song, Jeremy Getman, Joe Ellis, Justin Mott, Stephanie Strassel, Martha Palmer, Teruko Mitamura, Marjorie Freedman, Heng Ji, et al. A comparison of event representations in deft. In *Proceedings of the Fourth Workshop on Events*, pages 27–36, 2016.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM, 2008.

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

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

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

Kevin Burton, Akshay Java, and Ian Soboroff. The icwsm 2009 spinn3r dataset. In *Third Annual*

Conference on Weblogs and Social Media (ICWSM 2009). AAAI, 2009.

Hongyun Cai, Yang Yang, Xuefei Li, and Zi Huang. What are popular: exploring twitter features for event detection, tracking and visualization. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 89–98. ACM, 2015.

Cornelia Caragea, Adrian Silvescu, and Andrea H Tapia. Identifying informative messages in disaster events using convolutional neural networks. In *International Conference on Information Systems for Crisis Response and Management*, pages 137–147, 2016.

Tommaso Caselli and Oana Inel. 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, August 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W18-4306`.

Betul Ceran, Ravi Karad, Steven Corman, and Hasan Davulcu. A hybrid model and memory based story classifier. In *Proceedings of the 3rd Workshop on Computational Models of Narrative*, pages 58–62, 2012.

Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 602–610. Association for Computational Linguistics, 2009.

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

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

Timothy Chklovski and Patrick Pantel. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, 2004a.

Timothy Chklovski and Patrick Pantel. Verbocean: Mining the web for fine-grained semantic verb

relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004b.

Prafulla Kumar Choubey and Ruihong Huang. A sequential model for classifying temporal relations between intra-sentence events. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1796–1802, 2017.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, 2019.

Jacob Cohen. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213, 1968.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, 2017.

Zeyu Dai and Ruihong Huang. 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, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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 Lan-*

guage Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, 2019.

Oxford English Dictionary. Oxford english dictionary online, 2007.

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

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

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

Joshua Eisenberg and Mark Finlayson. A simpler and more generalizable story detector using verb and character features. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2698–2705, 2017.

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

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874, 2008.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

Edward Morgan Forster. Aspects of the novel. 1927. *Ed. Oliver Stallybrass*, 1962.

Lea Frermann, Ivan Titov, and Manfred Pinkal. A hierarchical bayesian model for unsupervised induction of script knowledge. In *EACL*, volume 14, pages 49–57, 2014.

Roxana Girju. 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, 2003.

Goran Glavaš and Jan Šnajder. 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, 2014.

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

Andrew Gordon and Reid Swanson. Identifying personal stories in millions of weblog entries. In *Third International Conference on Weblogs and Social Media, Data Challenge Workshop, San Jose, CA*, volume 46, 2009.

Hebert Grabes. Sequentiality. *Handbook of Narratology*, 2:765–76, 2013.

David Graff and C Cieri. English gigaword corpus. *Linguistic Data Consortium*, 2003.

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

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005.

Algirdas Julien Greimas. Narrative grammar: Units and levels. *MLN*, 86(6):793–806, 1971.

Catherine Havasi, Robert Speer, and Jason Alonso. Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent advances in natural language processing*, pages 27–29. Citeseer, 2007.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving Neural Networks by Preventing Co-adaptation of Feature Detectors. In *arXiv preprint arXiv:1207.0580*, 2012.

Zhichao Hu and Marilyn Walker. Inferring narrative causality between event pairs in films. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 342–351, 2017.

Zhichao Hu, Elahe Rahimtoroghi, Larissa Munishkina, Reid Swanson, and Marilyn A Walker. Unsupervised induction of contingent event pairs from film scenes. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 369–379, 2013.

Qunying Huang and Yu Xiao. Geographic situational awareness: mining tweets for disaster preparedness, emergency response, impact, and recovery. *ISPRS International Journal of Geo-Information*, 4(3):1549–1568, 2015.

Jagadeesh Jagarlamudi, Hal Daumé III, and Raghavendra Udupa. Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 204–213. Association for Computational Linguistics, 2012.

Bram Jans, Steven Bethard, Ivan Vulić, and Marie Francine Moens. 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, 2012.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of 2014 the Conference on Empirical Methods in Natural Language Processing (EMNLP-2014)*, 2014.

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

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916, 2013.

Amanda Lenhart. *Bloggers: A portrait of the internet's new storytellers*. Pew Internet & American Life Project, 2006.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation

embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

Hector Llorens, Estela Saquete, and Borja Navarro. 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, 2010.

Inderjeet Mani. Computational modeling of narrative. *Synthesis Lectures on Human Language Technologies*, 5(3):1–142, 2012.

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

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

Adam Marcus, Michael S Bernstein, Osama Badar, David R Karger, Samuel Madden, and Robert C Miller. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 227–236. ACM, 2011.

Tara McIntosh and James R Curran. Reducing semantic drift with bagging and distributional similarity. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 396–404. Association for Computational Linguistics, 2009.

P. Meladianos, G. Nikolentzos, F. Rousseau, Y. Stavrakas, and M. Vazirgiannis. Degeneracy-based real-time sub-event detection in twitter stream. In *Proceedings of the 9th AAAI international conference on web and social media (ICWSM)*, pages 248–257, 2015.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information*

processing systems, pages 3111–3119, 2013b.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013c.

Tomáš Mikolov, Édouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

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

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

Paramita Mirza and Sara Tonelli. Classifying temporal relations with simple features. In *EACL*, volume 14, pages 308–317, 2014b.

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

Teruko Mitamura, Yukari Yamakawa, Susan Holm, Zhiyi Song, Ann Bies, Seth Kulick, and Stephanie Strassel. Event nugget annotation: Processes and issues. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 66–76, 2015.

Ashutosh Modi and Ivan Titov. Inducing neural models of script knowledge. In *CoNLL*, volume 14, pages 49–57, 2014.

Ashutosh Modi, Tatjana Anikina, Simon Ostermann, and Manfred Pinkal. Inscript: Narrative texts annotated with script information. In *LREC*, pages 3485–3493, 2016.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North Ameri-*

can Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 839–849, San Diego, California, June 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N16-1098.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 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, 2012.

National Hurricane Center. Costliest u.s. tropical cyclones tables updated. Technical report, 2017. URL https://www.nhc.noaa.gov/news/UpdatedCostliest.pdf.

Roberto Navigli and Mirella Lapata. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE transactions on pattern analysis and machine intelligence*, 32(4):678–692, 2010.

Dat Tien Nguyen, Kamela Ali Al Mannai, Shafiq Joty, Hassan Sajjad, Muhammad Imran, and Prasenjit Mitra. Robust classification of crisis-related data on social networks using convolutional neural networks. In *Eleventh International AAAI Conference on Web and Social Media*, 2017.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 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, 2018.

Tim O'Gorman, Kristin Wright-Bettner, and Martha Palmer. 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, 2016.

Tim O'Gorman, Kristin Wright-Bettner, and Martha Palmer. 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, 2016.

James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. The development and psychometric properties of liwc2015. Technical report, 2015.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word

representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

Brian T Pentland. Building process theory with narrative: From description to explanation. *Academy of management Review*, 24(4):711–724, 1999.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, 2018.

Karl Pichotta and Raymond J Mooney. Learning statistical scripts with lstm recurrent neural networks. In *AAAI*, pages 2800–2806, 2016.

Daniela Pohl, Abdelhamid Bouchachia, and Hermann Hellwagner. Automatic sub-event detection in emergency management using social media. In *Proceedings of the 21st International Conference on World Wide Web*, pages 683–686. ACM, 2012.

Ana-Maria Popescu, Marco Pennacchiotti, and Deepa Paranjpe. Extracting events and event descriptions from twitter. In *Proceedings of the 20th international conference companion on World wide web*, pages 105–106. ACM, 2011.

Rashmi Prasad, Eleni Miltsakaki, Nikhil Dinesh, Alan Lee, Aravind Joshi, Livio Robaldo, and Bonnie L Webber. The Penn Discourse Treebank 2.0. In *Proceedings of the Sixth Conference on Language Resources and Evaluation (LREC-2008)*, 2008.

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

James Pustejovsky, Marc Verhagen, Roser Saurí, Jessica Littman, Robert Gaizauskas, Graham Katz, Inderjeet Mani, Robert Knippen, and Andrea Setzer. Timebank 1.2. *Linguistic Data Consortium*, 40, 2006.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A Smith, and Yejin Choi. Event2mind: Commonsense inference on events, intents, and reactions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 463–473, 2018.

Michaela Regneri, Alexander Koller, and Manfred Pinkal. Learning script knowledge with web experiments. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 979–988. Association for Computational Linguistics, 2010.

Mehwish Riaz and Roxana Girju. 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, 2010.

Mehwish Riaz and Roxana Girju. 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, 2013.

Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer, 2010.

Mark O Riedl and Robert Michael Young. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39:217–268, 2010.

E. Riloff, J. Wiebe, and T. Wilson. Learning Subjective Nouns using Extraction Pattern Bootstrapping. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*, pages 25–32, 2003.

Benjamin Roth, Tassilo Barth, Michael Wiegand, and Dietrich Klakow. A survey of noise reduction methods for distant supervision. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 73–78, 2013.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

Jagan Sankaranarayanan, Hanan Samet, Benjamin E Teitler, Michael D Lieberman, and Jon Sperling. Twitterstand: news in tweets. In *Proceedings of the 17th acm sigspatial international conference on advances in geographic information systems*, pages 42–51. ACM, 2009.

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

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205, 2006.

Chao Shen, Fei Liu, Fuliang Weng, and Tao Li. A participant-based approach for event summarization using twitter streams. In *HLT-NAACL*, pages 1152–1162, 2013.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 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, 2015.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Stephanie Strassel, Mark A Przybocki, Kay Peterson, Zhiyi Song, and Kazuaki Maeda. Linguistic Resources and Evaluation Techniques for Evaluation of Cross-Document Automatic Content Extraction. In *Proceedings of the Sixth International Language Resources and Evaluation Conference (LREC-08)*, 2008.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 721–729, 2012.

Niket Tandon, Gerard De Melo, Abir De, and Gerhard Weikum. Knowlywood: Mining activ-

ity knowledge from hollywood narratives. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 223–232, 2015.

Inc. Twitter. Gnip api, 2019. URL `http://support.gnip.com/apis/`.

Naushad UzZaman, Hector Llorens, James Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. SemEval-2013 task 1: TempEval-3 evaluating time expressions, events, and temporal relations. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 75–80. Association for Computational Linguistics, 2007.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 57–62. Association for Computational Linguistics, 2010.

Richard Walsh. Fabula and fictionality in narrative theory. *Style*, 35(4):592–606, 2001.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12): 2724–2743, 2017a.

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

Zhongqing Wang, Yue Zhang, and Ching-Yun Chang. 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, 2017b.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. Probase: A probabilistic taxonomy

for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 481–492. ACM, 2012.

Jierui Xie, Boleslaw K Szymanski, and Xiaoming Liu. 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, 2011.

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

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019.

Limin Yao, Sebastian Riedel, and Andrew McCallum. Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1013–1023, 2010.

Wenlin Yao and Ruihong Huang. Temporal event knowledge acquisition via identifying narratives. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 537–547, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1050. URL https://www.aclweb.org/anthology/P18-1050.

Wenlin Yao, Saipravallika Nettyam, and Ruihong Huang. A weakly supervised approach to train temporal relation classifiers and acquire regular event pairs simultaneously. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 803–812, Varna, Bulgaria, September 2017. INCOMA Ltd. doi: 10.26615/978-954-452-049-6_103. URL https://doi.org/10.26615/978-954-452-049-6_103.

Wenlin Yao, Zeyu Dai, Maitreyi Ramaswamy, Bonan Min, and Ruihong Huang. Weakly supervised subevent knowledge acquisition. In *Proceedings of the 2020 Conference on Empirical*

Methods in Natural Language Processing (EMNLP), pages 5345–5356, 2020a.

Wenlin Yao, Cheng Zhang, Shiva Saravanan, Ruihong Huang, and Ali Mostafavi. Weakly-supervised fine-grained event recognition on social media texts for disaster management. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 532–539, 2020b.

David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 189–196. Association for Computational Linguistics, 1995.

Wen-tau Yih, Xiaodong He, and Christopher Meek. Semantic parsing for single-relation question answering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014.

Cheng Zhang, Chao Fan, Wenlin Yao, Xia Hu, and Ali Mostafavi. Social media for intelligent public information and warning in disasters: An interdisciplinary review. *International Journal of Information Management*, 49:190–207, 2019.

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. Aser: A large-scale eventuality knowledge graph. In *Proceedings of The Web Conference 2020*, pages 201–211, 2020.

Deyu Zhou, Liangyu Chen, and Yulan He. An unsupervised framework of exploring events on twitter: Filtering, extraction and categorization. In *AAAI*, pages 2468–2475, 2015.

Zhi-Hua Zhou and Ming Li. Semi-supervised regression with co-training. In *IJCAI*, volume 5, pages 908–913, 2005.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

APPENDIX A

THE FULL LIST OF GRAMMAR RULES FOR IDENTIFYING PLOT EVENTS IN THE SEEDING STAGE OF NARRATIVE IDENTIFICATION

Here is the full list of grammar rules for identifying plot events in the seeding stage (Section 4.2.1).

Sentence rules (14):

S → S CC S

S → S PRN CC S

S → NP VP

S → NP ADVP VP

S → NP VP ADVP

S → CC NP VP

S → PP NP VP

S → NP PP VP

S → PP NP ADVP VP

S → ADVP S NP VP

S → ADVP NP VP

S → SBAR NP VP

S → SBAR ADVP NP VP

S → CC ADVP NP VP

Noun Phrase rules (12):

NP → PRP

NP → NNP

NP → NNS

NP → NNP NNP

NP → NNP CC NNP

NP → NP CC NP

NP → DT NN

NP → DT NNS

NP → DT NNP

NP → DT NNPS

NP → NP NNP

NP → NP NNP NNP

APPENDIX B

THE FULL LIST OF KEYWORDS USED FOR EACH EVENT CATEGORY IN FINE-GRAINED EVENT DETECTION ON SOCIAL MEDIA

Here is the full list of keywords used for each event category (Section *Event Categories and Event Keywords*). Various word forms of the keywords are also considered, e.g., "evacuates, evacuated, evacuating" are also considered for the keyword "evacuate".

1) Preventative measure (PRE): evacuate, evacuation, evacuee, shelter, refugee

2) Help and rescue (RES): rescue, boat, help, donate, guard

3) Casualty (CAS): die, dead, drown, injure, hurt

4) Housing (HOU): house, home, room, apt, apartment

5) Utilities and Supplies (UTI): power, electricity, gas, store, food, supply

6) Transportation (TRA): airplane, plane, flight, airport, "RoadTypes" (highway, freeway, road, avenue, ave, dr, rd, st, hwy, fwy, blvd)

7) Flood control infrastructures (FCI): reservoir, bayou, canal, dam, levee

8) Business Work School (BWS): office, school, closed, open, work

9) Built-environment hazards (HAZ): fire, explosion, collapse, poison, electrocute