Event Detection and Encoding from News Articles

Wei Wang

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

Narendran Ramakrishnan, Chair
Chang Tien Lu
Christopher North
Huzefa Rangwala
Ravi Tandon

September 11, 2017
Arlington, Virginia

Keywords: Event Detection, Event Encoding, Deep Learning, Multi Instance Learning,
Multi Task Learning

Event Detection and Encoding from News Articles

Wei Wang

(ABSTRACT)

Event extraction is a type of information extraction(IE) that works on extracting the specific knowledge of certain incidents from texts. Nowadays the amount of available information (such as news, blogs, and social media) grows in exponential order. Therefore, it becomes imperative to develop algorithms that automatically extract the machine-readable information from large volumes of text data. In this dissertation, we focus on three problems in obtaining event-related information from news articles. (1) The first effort is to comprehensively analyze the performance and challenges in current large-scale event encoding systems. (2) The second problem involves event detection and critical information extractions from news articles. (3) Third, the efforts concentrate on event-encoding which aims to extract event extent and arguments from texts.

We start by investigating the two large-scale event extraction systems (ICEWS and GDELT) in the political science domain. We design a set of experiments to evaluate the quality of the extracted events from the two target systems, in terms of reliability and correctness. The results show that there exist significant discrepancies between the outputs of automated systems and hand-coded system and the accuracy of both systems are far away from satisfying. These findings provide preliminary background and set the foundation for using advanced machine learning algorithms for event related information extraction.

Inspired by the successful application of deep learning in Natural Language Processing (NLP), we propose a Multi-Instance Convolutional Neural Network (MI-CNN) model for event detection and critical sentences extraction without sentence level labels. To evaluate the model, we run a set of experiments on a real-world protest event dataset. The result shows that our model could be able to outperform the strong baseline models and extract

the meaningful key sentences without domain knowledge and manually designed features.

We also extend the MI-CNN model and propose an MIMTRNN model for event extraction with distant supervision to overcome the problem of lacking fine level labels and small size training data. The proposed MIMTRNN model systematically integrates the RNN, Multi-Instance Learning, and Multi-Task Learning into a unified framework. The RNN module aims to encode into the representation of entity mentions the sequential information as well as the dependencies between event arguments, which are very useful in the event extraction task. The Multi-Instance Learning paradigm makes the system does not require the precise labels in entity mention level and make it perfect to work together with distant supervision for event extraction. And the Multi-Task Learning module in our approach is designed to alleviate the potential overfitting problem caused by the relatively small size of training data. The results of the experiments on two real-world datasets(Cyber-Attack and Civil Unrest) show that our model could be able to benefit from the advantage of each component and outperform other baseline methods significantly.

Event Detection and Encoding from News Articles

Wei Wang

(GENERAL AUDIENCE ABSTRACT)

Nowadays the amount of available information (such as news, blogs, and social media) grows in exponential order. The demand of making use of the massive on-line information during decision making process becomes increasing intensive. Therefore, it is imperative to develop algorithms that automatically extract the formatted information from large volumes of the unstructured text data. In this dissertation, we focus on three problems in obtaining event-related information from news articles. (1) The first effort is to comprehensively analyze the performance and challenges in current large-scale event encoding systems. (2) The second problem involves detecting the event and extracting key information about the event in the article. (3) Third, the efforts concentrate on extracting the arguments of the event from the text. We found that there exist significant discrepancies between the outputs of automated systems and hand-coded system and the accuracy of current event extraction systems are far away from satisfying. These findings provide preliminary background and set the foundation for using advanced machine learning algorithms for event related information extraction. Our experiments on two real-world event extraction tasks (Cyber-Attack and Civil Unrest) show the effectiveness of our deep learning approaches for detecting and extracting the event information from unstructured text data.

Acknowledgments

First and foremost I would like to express my sincere appreciation to my wonderful advisor Prof. Naren Ramakrishnan for the continuous support of my Ph.D. study and research. I would like to thank you for your patience, ideas, and fundings to make my Ph.D. experience productive and stimulative. You provided me not only the guidance on research but also the wisdom of life which helped me get through tough times in the Ph.D. pursuit. Your advice on both research and life have been invaluable to me.

I would also have to thank my committee members: Prof. Chang-Tien Lu, Prof. Chris North, Prof. Huzefa Rangwala, and Prof. Ravi Tandon, for their brilliant comments, and suggestions. Special thanks to Prof. Huzefa Rangwala, without your insightful guidance and advice, I could not make it as much as I have.

I would also like to thank my colleagues in DAC: Dr. Hao Wu, Dr. Fang Jin, Dr. Huijuan Shao, Yue Ning, Rongrong Tao, Sathappan Muthiah, Rupinder Paul, Parang Saraf, Saurav Ghosh, Mohammad Raihanul Islam, Malay Chakrabarti, Yaser Keneshloo, Nikhil Muralidhar, Siddharth Krishnan, Dr. Prithwish Chakraborty, Dr. K.S.M. Tozammel Hossain, Dr. Patrick Butler, Dr. Brian Goode, Nathan Self, Peter Hauk, Juanita Victoria, Wanawsha Hawrami, and Joyce Newberry. Thanks all for making DAC such a great place.

Finally and most importantly, my deep and sincere gratitude to my family for their unparalleled love and support. I am especially grateful to my parents, thank you for raising me and inspiring me to follow my dreams. I would also like to thank my beloved wife, Ying Ni. Without your selfless support and encouragement, the journey would not have been possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the exponentially increasing amount of available text data such as news, blogs and social media, it has become an urgent and critical matter to utilize the information from these data in decision making process. The ability of ingesting large amount of information from multiple sources could alleviate the problem of source bias and enable a quick response to the emerging changes. However, a ubiquitous problem is that the most of the text data are initially unstructured and are represented using natural languages. The form of natural language is challenging to the machine's ability to read and understand the information properly. On the other hand, human mind is limited by the amount of information that it can process. As a result, automatic extraction of specific information from free texts has become a popular area of research. By the means of Natural Language Process (NLP) and Machine Learning (ML), the research aims to extract the knowledge from unstructured text data and represent it in a structured way such as database record. As a special type of IE tasks, event extraction targets structured incident information, which can be represented by a combination of complex relations linked by entities from texts.

Here is an specific example of an event record. In a tuple (< Attacker[Organization]>, <Attack>, <Target[Organization]>) that represents a cyber attack event, the organization entities in the text might be assigned with the role of $< Attacker >$ or $< Target >$, and the words describing cyber attack are associated with $< Attack >$. One example of text which matches the event representation could be "*The hacktivist* ***Anonymous*** *has* ***taken down*** *the website of* ***Microsoft Xbox***."

Event extraction from texts could be beneficial to various domains. In news aggregation application, the ability of determining events could enhance the performance of personal news recommendation, since the news articles could be selected more accurately based on the extracted events and user preference. For government policy makers, having the ability of tracking the significant international incidents efficiently could help to make better decisions. In cyber security domain, experts could be able to adapt to the rapidly evolving threats by monitoring the ongoing cyber attack incidents. Another possible application of event extraction lies in the area of algorithmic trading, which utilizes computer algorithms to decide the stock trading strategies such as time, price and volume. The stock market is extremely sensitive to the breaking news, thus efficiently extracting the market-related information from text could help the system to respond quickly to the emerging events.

A lot of efforts have been devoted to the event extraction area in the recent past. We decompose prior works into two interrelated subproblems: (1) event detection (or recognition) – identification of the documents describing a specific event; (2) event encoding (or extraction) – identification of the phrases, tokens or sentences (with relationships) that provide detailed information about the event e.g., type of event, location of event, people involved, and time of the event. Event detection and encoding pose a multitude of challenges due to the variety of event domains, types, definitions and expectations of the algorithms.

In general, the efforts on event encoding (extraction) can be categorized into two groups: open information extraction and domain-specific event extraction. Open information extraction methods normally take text as input and output tuples that include two entities and the relationship between them (e.g., Teachers (entity), government (entity), protest against (relationship)). Domain-specific event extraction approaches rely on templates, dictionaries, or presence of a specific structure within the input text. These input templates of events vary dramatically based on different situations. For instance, an earthquake event template might contain location, magnitude, number of missing people, damage to the infrastructure, and time of the event. Whereas, a civil unrest event template might contain fields like participants, purpose, location, and time. Most prior event extraction research [21, 86, 125] has focused on extracting entities, detecting trigger terms (or keywords), and matching up event slots on predefined templates. For example, Huang et. al. [50] propose a bootstrapping approach to learn event phrase, agent term, and purpose phrase for event recognition. Entity-driven probabilistic graphical models [21, 86, 125] were proposed to jointly learn the

event templates and align the template slots to identified tokens. This study focuses on the domain-specific event extraction.

1.1 Goals of the Dissertation

Given the huge potential of the applications of event extraction and current rapid development of Natural Language Processing and Machine Learning technologies, it is worth to re-investigate the current event encoding systems and enhance the event extraction performance with the aid of the emerging technologies. In this dissertation, we aim to solve following three research questions:

RQ1: Large-Scale Event Extraction System Evaluation

In the past decades, a great amount of research efforts has been devoted to text understanding and information extraction. In the area of political science, two well-known event encoding systems ICEWS and GDELT have been developed to automatically extract the international political incidents such as protests, assaults and mass violence from news media. However, very little work has been done to comprehensively evaluate the quality of the extracted information from the large-scale event encoding systems. It is anticipated that the in-depth analysis of the encoded events could help to find the defects of current systems and to propose new algorithms to overcome these challenges. How can we systematize the evaluation of the current event encoding systems?

RQ2: Event Detection and Key sentences Extraction

Identifying and extracting relevant information from the large volumes of text play a critical role in various applications. For the task of event detection, we need to identify the documents which describe a specific event. In addition to detecting the occurrence of a specific incident, it is also important to extract the sentences which provide detailed information about the event. The challenge is that it is extremely costly and time-consuming to manually label the articles at a sentence level. However, the labels at an article level are relatively easier to obtain. How could we train a system to complete the two tasks (event detection and key information extraction) simultaneously without the sentence level labeling?

RQ3: Event Encoding

Besides the event detection, event encoding is another task which aims to extract the event arguments from an article which describes a specific event. Similar to the challenges faced by key-sentence extraction task, it is challenging to annotate the event arguments at a token or phrase level. Different to traditional classification problems, there is a strong correlation between the labels of event arguments in the text. For instance, generally it would be more possible to label two entity mentioning in one sentence as Attacker and Target than as Attacker and Attacker. Previous works [21, 20, 69] in event extraction often ignore this kind of dependency and classify each event argument candidate independently, while our proposed approach utilizes the Recurrent Neural Network to encode this dependency into entity mention embedding explicitly. Furthermore, the tasks of event detection and argument classification are also strongly correlated. Thus, our final research question is: How can we collectively detect an event and extract the event arguments?

1.2 Organization of the Dissertation

The remaining dissertation proposal is organized as follows.

In Chapter 2, we will introduce the background information and some preliminaries that will be useful to understand the technologies in our approaches. This chapter will cover some basic concepts in feedforward Neural Network, word2vec, CNN, RNN and multi-instance learning.

In Chapter 3, we investigate the performance of two well-known large-scale political event encoding systems GDELT and ICEWS from both reliability and validity point of view. Our study shows that correlations between GSR, GDELT and ICEWS are relatively weak at all levels: daily, weekly, and monthly. The manual check of the GDELT dataset shows that the quality of the system is not satisfying and the average accuracy is only around 16.2%.

In Chapter 4, the problem of event detection and key information extraction is addressed. Specifically, a Deep Multi-Instance framework was designed to transfer the article level label to sentence level. The results of experiments on the real-world protest event dataset show that our approach could outperform the strong baseline methods and extract the meaningful event extents from the articles.

In Chapter 5, we study the problem of event encoding with distant supervision. We proposed an MIMTRNN model which systematically integrate the RNN, multi-instance learning, and multi-task learning into a unified framework. Furthermore, we analyze the impact of each component in the model by a set of experiments on a real-world cyber attack event dataset.

Finally, in Chapter 6, we offer some concluding remarks as well as some thoughts on future work.

Chapter 2

Preliminaries

In this chapter, we will introduce the background information of the technologies used in our approaches. It would include preliminaries for deep learning and multi-instance learning.

2.1 Deep Learning Background

In this section, we will review the basic form of neural network model, explain how words are represented in deep learning, and introduce two widely used deep learning modes (CNN and RNN) in Natural Language Processing.

Feedforward Neural Network

We begin by introducing the most basic form of neural network model, namely the feedforward neural network. Figure 2.1 shows a four-layers feedforward neural network with two hidden layers. Each unit in the network is called *neurons*, and there are 7 *input neurons* in the network shown in figure 2.1. Given the input x, weight matrix W, bias vector b, and activation function f, the layer output a is computed by:

$$z = W^T \cdot x + b \tag{2.1}$$

$$a = f(z) \tag{2.2}$$

We usually call z as pre-activation and activation function f as nonlinearity, which often

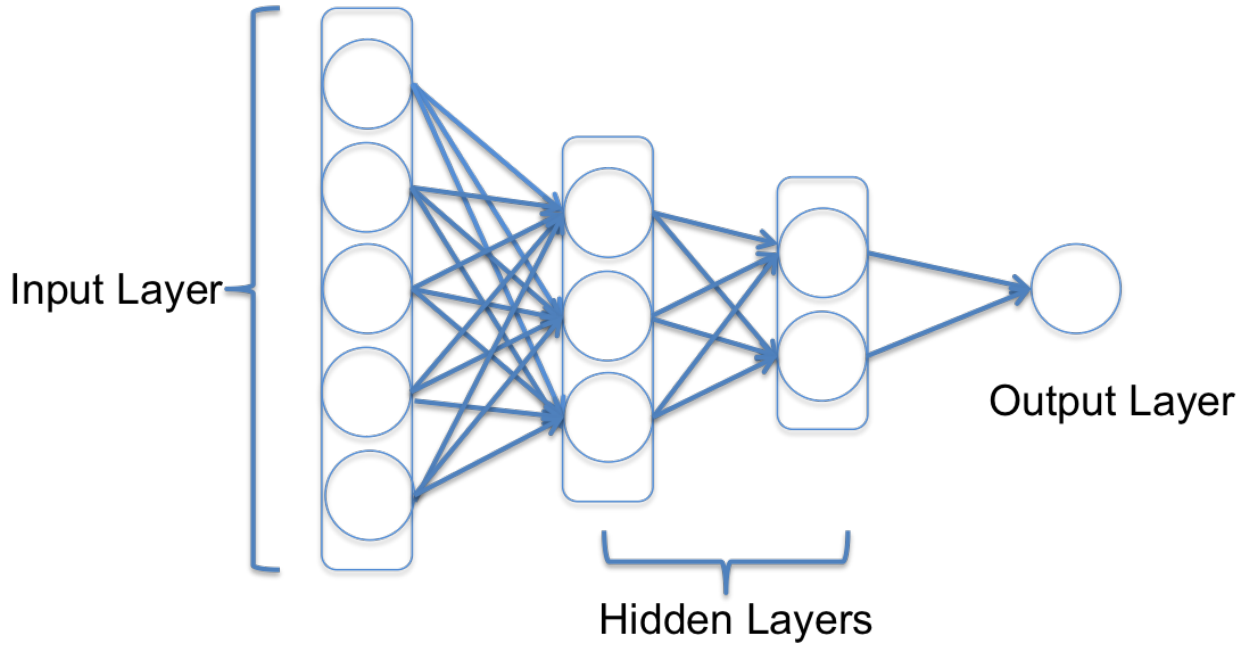

Figure 2.1: A example of four layers feedforward neural network

uses the Sigmoid function:

$$f(x) = sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

or the Rectifier function.

$$f(x) = Rectifier(x) = max(0, x) \tag{2.4}$$

We use superscript to denote the different layer when multiple layers are stacked over each other.

$$z^{\ell} = W^{\ell} \cdot a^{\ell-1} + b^{\ell} \tag{2.5}$$

$$a^{\ell} = f(z^{\ell}) \tag{2.6}$$

In the last layer L, the output o is usually computed by applying a softmax function the classification problem:

$$o_i = softmax(z_i^L) = \frac{e^{z_i^L}}{\sum_j e^{z_j^L}} \tag{2.7}$$

or a linear operation for regression problem. Finally, a minimized error function would be applied on the last layer's output o. The most common used error function for classification problem is cross-entropy loss:

$$C = -\frac{1}{N}\sum_{n}\sum_{i} y_{ni} log(o_{ni}) \tag{2.8}$$

or least square error for regression problem:

$$C = \frac{1}{N}\sum_{n}(o_n - y_n)^2 \tag{2.9}$$

here y_n is the ground truth for the n_{th} training example x_n.

The standard learning algorithm in the neural network is stochastic gradient descent (SGD). The pseudocode for the learning process of SGD can be presented as below:

Initialize the parameter vector w and learning rate η;
while *until the stop condition meet* **do**
  Randomly shuffle the examples x in the training set;
  for *i = 1 to n* **do**
    compute the loss c for x_i;
    compute the gradident $\Delta w := \nabla c(w)$;
    update the parameter $w := w - \eta \Delta w$;
  end
end

Algorithm 1: Pseudocode for SGD algorithm

Today, *backpropagation* [102] is the most common used algorithm to compute the gradients in the neural network. It can make training with gradient descent much faster than a native implementation, and make the training of deep models computationally feasible. Essentially, *backpropagation* algorithm is an application of chain rule for calculating derivative. The core of *backpropagation* algorithm in a feedforward neural network could be built on four foundational equations [2.10, 2.11, 2.12, 2.13]. Let's define the error in layer ℓ as δ^ℓ and the loss as C. Then an equation of the error ℓ^L for the output layer L is:

$$\delta^L = \frac{\partial C}{\partial a_L} f'(z^L) \tag{2.10}$$

The equation of the error in middle layer ℓ is:

$$\delta^\ell = (W^{\ell+1})^T \cdot \delta^\ell \odot f'(z^\ell) \tag{2.11}$$

Given the equations 2.10 and 2.11, we could compute the error for any layer, all the way back through the network. The equation for the gradient with respect to the bias b in layer ℓ is:

$$\frac{\partial C}{\partial b^{\ell}} = \delta^{\ell} \tag{2.12}$$

And the equation for the gradient with respect to the weight W is given by:

$$\frac{\partial C}{\partial W^{\ell}} = \delta^{\ell} \cdot (a^{\ell-1})^{T} \tag{2.13}$$

Word2Vec

In traditional language processing algorithms, the words are usually represented by the "one hot" representation, which has a single one at the position of the word index and all 0s at other positions. The problem of "one hot" representation is that it can not capture the context information. For instance, the word *dog* and *puppy* synonyms for each other, while the similarity between the "one hot" representations can not reflect this information.

In word2vec, each word is represented by a densely distributed representation. So instead of one to one mapping between the elements in the vector and a word, each word is represented by all the elements in the vector. The idea behinds the word2vec is that one word could be described by its neighbors. Firth [36] first introduced this idea in 1957 and it has been used in the domain of NLP extensively. For instance, the Latent Dirichlet Allocation(LDA) could be considered as using a whole document as the context for the word in that document.

Bengio et al. [15] proposed a neural network language model which learned the distributed representations for words with context-window information at the same time. Mikolov et al. [80] proposed an efficient way to learn the distributed representation of words and make it practical to apply on the large corpus. There are two architectures being proposed in [80] : Continuous Bag Of Words (CBOW) and Continuous Skip-Gram. The CBOW model predicts the current word based on the context, while the Skip-Gram model predicts surrounding words given the current word.

The word2vec representation has shown several interesting properties. First, it could be able to capture the semantic information, the words with similar context usually has close representations in Euclidean space. Moreover, as shown in [80], there exist some linear rela-

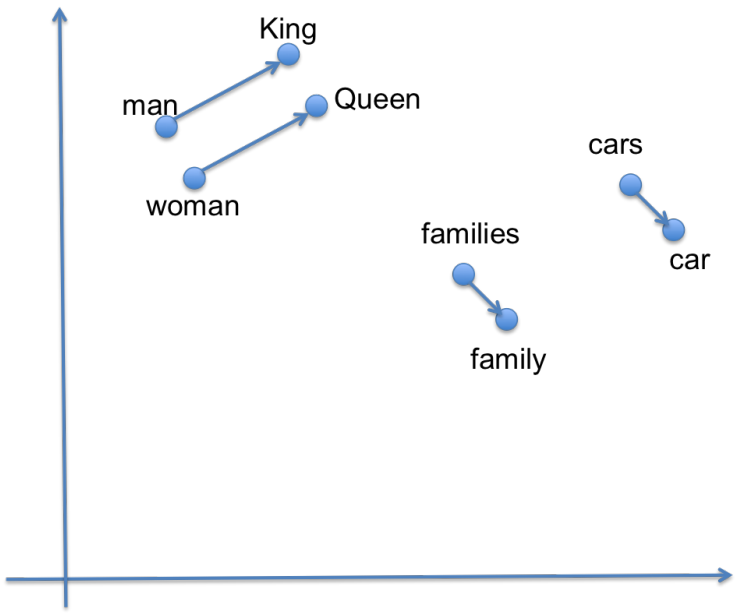

Figure 2.2: Demonstration of the linear relations between word2vec: vec(King) - vec(man) ≈ vec(Queen) - vec(woman) and vec(families) - vec(family) ≈ vec(cars) - vec(car)

tions between the learned representations. For instance, as shown in Figure 2.2, *vec(King) - vec(man) ≈ vec(Queen) - vec(woman) and vec(families) - vec(family) ≈ vec(cars) - vec(car).*

Convolutional Neural Network

The architecture of Convolutional Neural Network(CNN) is similar to the aforementioned feedforward neural network. They are both built on the forward stacked layers which are consisted of nonlinear neurons. The difference is that in feedforward neural network each layer is fully connected to the previous and following layers, while CNN consists of some convolutional layers which are followed by a subsampling layer. This difference makes CNN more powerful at capturing the spatial dependencies. Figure 2.3 shows an architecture of the Convolutional Neural Network.

The convolutional layer is the core block to build CNN, and its parameters consist of a set of learnable filters. Unlike the hidden units which are computed based on all the units in the previous layer in feedforward neural network, the neuron in the convolutional layer is only connected to a local region in the previous layer. The output of the neuron is computed by the dot product of the filter and the connected local area followed by a nonlinear activation.

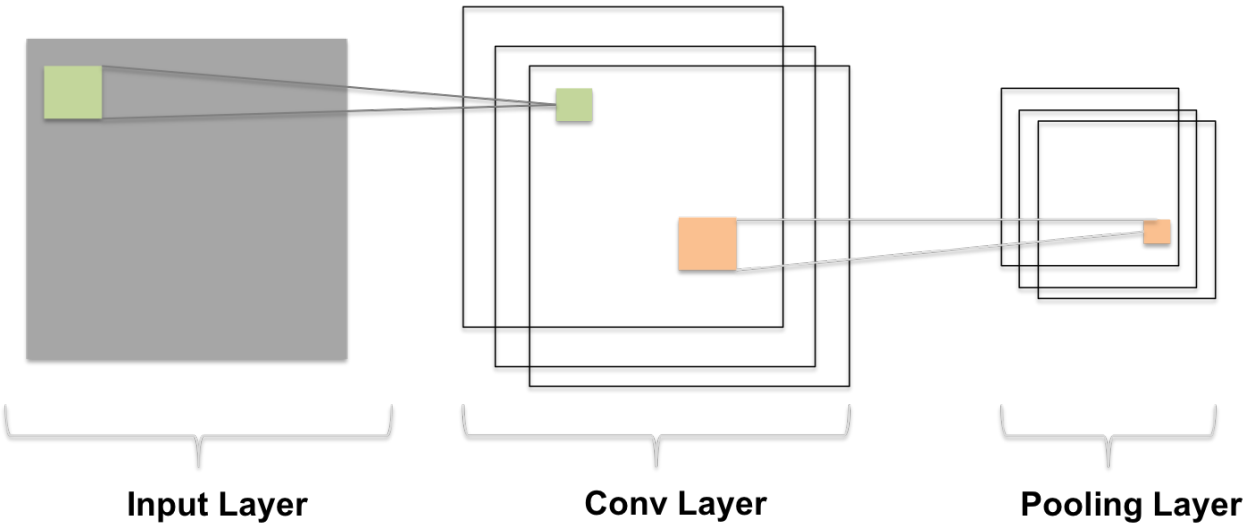

Figure 2.3: Architecture of Convolutional Neural Network (not including the output layer)

Given a filter parameter $W \in R^{m \times m}$ and a location region input X, and the activation function f, the equation for computing the output o of a Conv neuron is:

$$o = f(\sum_{a=0}^{m-1} \sum_{b=0}^{m-1} W_{ab} \cdot X_{ab}) \tag{2.14}$$

Each filter would scan through the whole input matrix, and the outputs constitute a feature map. The parameter sharing in feature map not only reduces the number of parameters dramatically, but also makes sense to detect the pattern features no matter where they are located.

After each convolutional layer, there might be a downsampling or pooling layer. Similar to the convolutional layer, the pooling layer takes a sliding window from the convolutional layer and subsamples it to a single value. There are several common ways to do the pooling operation, such as maximum, average or linear combination.

Recurrent Neural Network

Recurrent Neural Network (RNN) is one particular type of neural network where the connections between its neurons form a directed circle. In contrast to the traditional feedforward network, RNN keeps track of its internal hidden state through recurrent connections. This behavior makes RNN suitable to process the tasks with sequence data such as text and

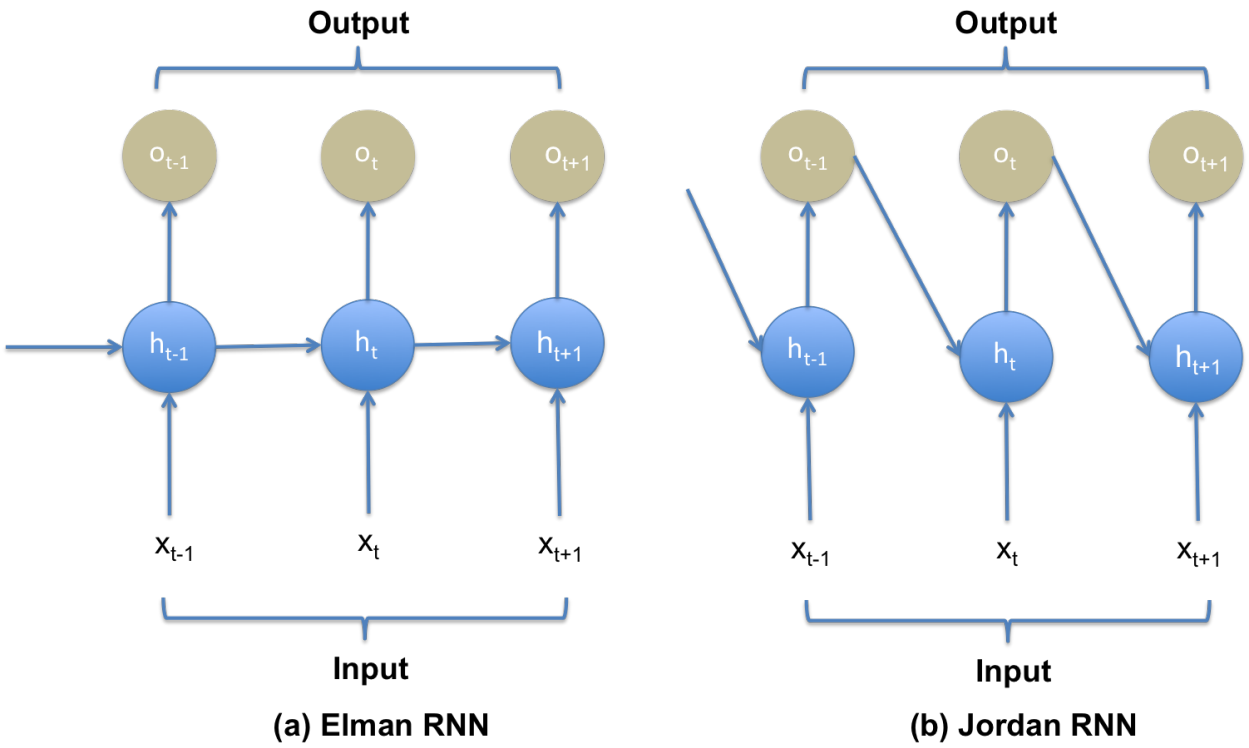

Figure 2.4: Two common RNN architectures

speech. The idea behinds the RNN is to utilize the sequential information. For instance, it's better to predict the next word in the sentence given all the words before that word.

We introduce two most common used RNN architectures in this section: the Elman RNN [34] and Jordan RNN [56]. Figure 2.4 shows the time unrolling version of Elman-RNN and Jordan-RNN. The difference between these two types of RNN architecture is that: in Elman RNN the hidden states connect to its previous state recursively, while in Jordan-RNN the hidden states rely on the previous output. Mathematically, the dynamics of hidden states in Elman RNN work as below:

$$h_t = f_h(W_h \cdot h_{t-1} + W_x \cdot x_t + b_h) \tag{2.15}$$

$$o_t = f_o(W_y \cdot h_t + b_o) \tag{2.16}$$

And in Jordan-RNN the process works as:

$$h_t = f_h(W_h \cdot o_{t-1} + W_x \cdot x_t + b_h) \tag{2.17}$$

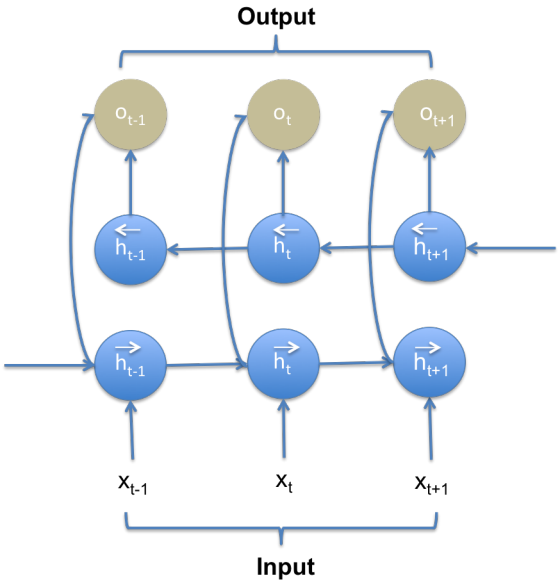

Figure 2.5: Architecture of bidirectional Elman RNN

$$o_t = f_o(W_y \cdot h_t + b_o) \tag{2.18}$$

Here,

- h_t: hidden state in time step t
- W_h, W_x, b_h, b_o, W_y: parameters
- o_t: output at time step t
- f_h: activation function for hidden state
- f_o: output function

In event extraction, the information in future is also useful. It's not necessary to only consider the sequence in a single forward pass. It is also possible to take into account future information with a backward pass. Bidirectional RNN follows this idea exactly, and figure 2.5 shows the architecture for a bidirectional Elman RNN.

Long Short Term Memory networks (LSTM) One of the appeals of RNN is that it can connect the previous information to current task. In theory, RNN is capable of handling arbitary long-term dependencies, while the vanilla RNN suffers from the problem

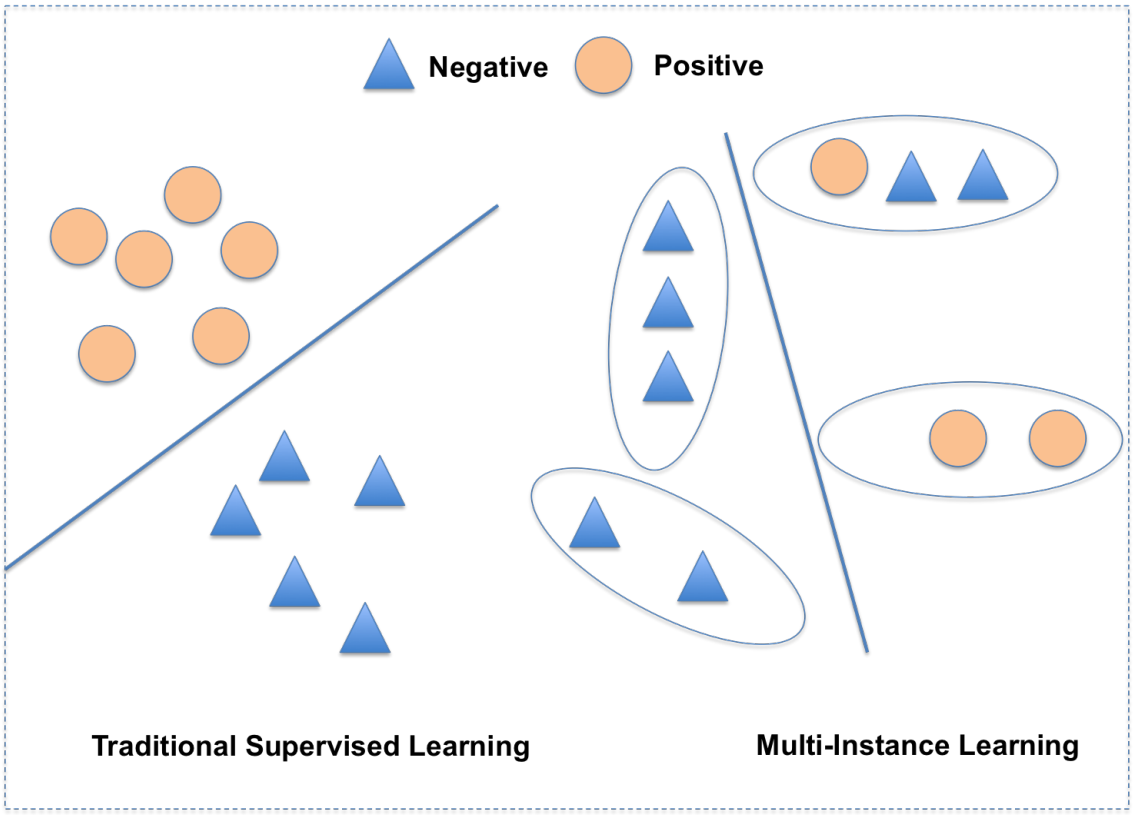

Figure 2.6: Comparing traditional supervised learning an multi-instance learning

of vanish/exploding gradient in practice when training the long sequence. LSTM network is a variant of RNN and it is much better at capturing the long-term dependencies than standard RNN.

The key of the LSTM is *Memory Block* which consists of three gates(Input, output and forget) and one cell state. The LSTM has the ability to remember or remove the information to the cell state. The gates are made of a sigmoid layer and a pointwise multiplication opertion. Each gate outputs a value between zero and one, describing how much information could pass through the gate. Figure 2.7 illustrates the an example of a birectinonal LSTM network structure. Mathmatically, the output of the LSTM cell h_t for input x_t at time step t is calculated through following equations:

$$InputGate : i_t = \sigma(W_i \cdot [x_t, h_{t-1}] + b_i) \tag{2.19}$$

$$ForgetGate : f_t = \sigma(W_f \cdot [x_t, h_{t-1}] + b_f) \tag{2.20}$$

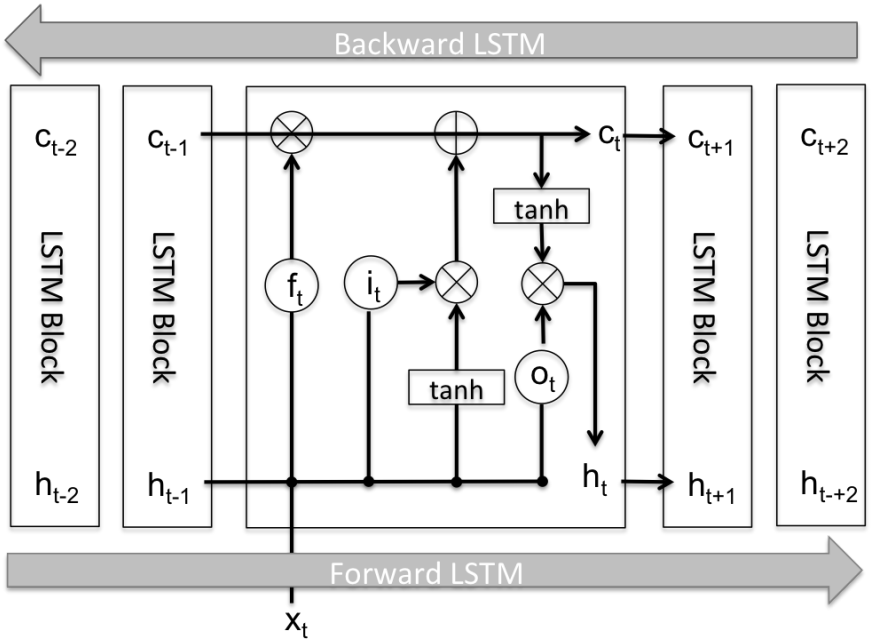

Figure 2.7: Example of a birectional LSTM network structure

$$OutputGate : o_t = \sigma(W_o \cdot [x_t, h_{t-1}] + b_o) \tag{2.21}$$

$$CandidateState : \hat{C}_t = tanh(W_c \cdot [x_t, h_{t-1}] + b_c) \tag{2.22}$$

$$CellState : C_t = f_t \star C_{t-1} + i_t \star \hat{C}_t \tag{2.23}$$

$$output : h_t = o_t \star tanh(C_t) \tag{2.24}$$

Here, W_i, W_f, W_o are gate coefficient matrix and b_i, b_f, b_o are bias parameters.

2.2 Learning Paradigms for Event Extraction

There are several learning paradigms(Supervised, Unsupervised, Semi-Supervised and Distant Supervision) have been applied to the task of Event Extraction.

Supervised Learning In supervised approaches, the sentences in the corpus have been manually labeled for the presence of the entities and their roles in the event. Labeling the training data is an expensive and time consuming process and thus limits the amount and quality of the dataset.

Unsupervised Learning An alternative approach to supervised approaches, unsupervised event extraction, works on extracting a set of entities based on some predefined rules(such as Semantic/Dependency Parsing) and then clustering them to produce event roles. Unsupervised learning approaches can use a very large amount of unlabeled text and generate a large number of events, while it is hard to map the extracted results to a specific event database.

Semi-Supervised Learning In semi-supervised learning, the methods can also approach to a large size of unlabeled corpus in addition to the small set of labeled instances. The labeled instances are used as seeds to do bootstrap learning. These seeds are used along with a large unlabeled corpus to generate more patterns in an iterative process. One problem of semi-supervised learning paradigm is that it may suffer from semantic drift and low precision.

Weak Supervision In weak supervision, the training dataset is noise and not perfectly accurate. The following examples can be thought as weak supervision:

- Domain heuristics: Generate training dataset according to domain knowledge such as common patterns and rule of thumb
- Existing databse: We only have labels on high level. For instance, any sentence that contains all entities of the event record might express that event in some way. This is traditionaly called **Distant Supervision**
- Noise Labels: The labels might be quite noise, for instance, labeled by non-experts.

2.3 Multi-Instance Learning

In this section, we will introduce the Multi-Instance Learning (MIL) paradigm. MIL is one type of supervised learning while handling the datasets which are more complicated than commonly presented. Figure 2.6 demonstrates a comparison between traditional supervised learning and Multi-Instance Learning. The traditional supervised learning could be called "single instance learning", since the observation used in traditional supervised learning is

usually a feature vector associated with an outcome. The learning object in MIL is called a *bag*, which consists of a set of instances represented by the feature vector. The outcome only associates with the bag, while the outcome in instance level is absent. The Multi-Instance Learning paradigm can be naturally applied in following situations [46]:

- **Distant Supervision:** Distant learning is a learning paradigm in which the learning algorithm is learned given a weakly labeled dataset. Multi-Instance learning could be used to handle noisy labels. For instance, in event extraction given an event record, it is natural to build each entity as a bag which consists of its mentions in the article and assign the event argument in entity level.
- **Compound Objects:** Each compound object consists of several parts, while only one or several parts of the object relate to the object label. For example, in an image with the label *human*, while only the segment where the person locates really contributes to the label.
- **Alternative Representations:** One object might have many alternative feature vectors (instances) to describe it, and only one or subset of these instances may be responsible for the observed object label. The task is to determine which representation leads to the final output. This is the original application of MIL in drug activity prediction [28].

Algorithm Taxonomy

There are several attempts [28, 121, 39, 7] to categorize the algorithms for Multi-Instance Learning. In this subsection, we will introduce the taxonomy proposed in [7].

- **Instance Space Method:** These type of algorithms work on predicting the label in instance space. Then the bag level labels are derived from the linking function over instances' labels. To infer the bag label without having access to the labeled instances, some assumptions must be made about the relation between bag label and instance label. In [7], two types of assumptions (Standard Multi-Instance assumption and Collective assumption) are introduced. The Standard Multi-Instance assumption states that every positive bag contains at least one positive instance, while all the instances

are negative in every negative bag. On the other hand, the Collective assumption states that all the instances contribute equally to the bag label. The Instance Space approaches might not work well if the inferring of bag label need the information beyond the single instance. The Diverse Density(DD) [77], Expectation-Maximization Diverse Density (EM-DD) [129], MI-SVM [8], Sparse MIL [17] algorithms fall into this category.

- **Bag Space Method:** These type of methods work in the bag space and try to determine the classes of the bag through the similarity between bags. In contrast to the Instance Space approaches which only consider the local information, the Bag Space approaches treat each bag as a whole. To define the similarity between bags, a distance function is usually defined to compare any two bags. Then the distance function can be plugged into any distance-based classifiers such as SVM and KNN. The common distance functions used in Bag Space approaches are the minimal Hausdorff distance [116], the Earth Movers Distance (EMD) [128] and the Chamfer distance [14].

- **Embedding Space Method:** These algorithms work in embedding space and try to map the entire bag to a vector space. Then the traditional single instance learning algorithms could be able to apply on the learning object with bag embedding and outcome. The mapping function could be a simple average or min/max function [31, 40] over all instances in the bag. Another type of mapping function is built by analyzing the pattern of instances in the bag [113, 106, 7]. The different types of mapping function could have a great impact on the performance of the method.

Chapter 3

Evaluation of Existing Event Encoding Systems

3.1 History of Event Data

In this section, we will attempt to give a succinct account of the development of event data over time. We will also discuss some of the legal issues surrounding GDELT and how they illustrate the need for action by government agencies to promote effective event data analysis. For a more complete analysis of the history of event data, we refer the interested reader to [104]

3.1.1 Early Event Coding

Scholars have long recognized that news reports were a goldmine of information on global events. Academic efforts to turn news reports into systematic event data date back to the 1960s and 1970s [10]. The U.S. Department of State and DARPA sponsored large-scale event data collection projects during this time [104]. These early projects were limited by their reliance on expensive human coding and a lack of long-term impact on the formulation of foreign policy. Because of these issues, all but a few event data collection efforts ceased in the 1980s. In the early 1990s, the rise of more powerful personal computers revived scholarly

and government interest in event data collection, this time through use of automated coding techniques. Most of these data collection efforts were limited in scale and focused on a particular region [104].

Moreover, while some prominent publications in political science came from these data collection efforts, the community involved in event data collection and analysis remained relatively small. As Schrodt and Gerner [104] discuss in the provenance of their online event data book, even after more than a decade of research into computer generated event data, it still suffered from a "lack of a commercially-viable audience". For most computer scientists, the concern of producing reliable event data was generally peripheral to the goals of the discipline, while, for most political scientists, the lack of training in computer programming was a barrier.

3.1.2 Current Event Coding Projects

Two recent data collection efforts have dramatically expanded the scope of news resources utilized and produced global event data coverage across a range of events. Moreover, these datasets are updated in near-real-time, making them useful for real-time conflict analysis. The Integrated Crisis Early Warning Systems (ICEWS), developed with support from DARPA, extended earlier event coding frameworks using a broad range of news resources [89]. ICEWS data is currently maintained by Lockheed-Martin and a portion of the data was recently released to the public on Harvard's Dataverse [3]. One of the issues with ICEWS is that the code utilized for generating the event data from the news sources is not open source.

The Global Database of Events, Language and Tone (GDELT) was built as a public and more transparent version of ICEWS. Its release in April 2013 was met with widespread enthusiasm [61, 45]. One of its authors was even named a top-100 global thinker by Foreign Policy magazine [2]. Unfortunately, controversy over how it obtained some of its news resources dampened academic interest in the project and resulted in several of the projectâĂŹs co-authors to distance themselves from the project. The data, however, continues to be utilized for analysis of international events [66, 65], is still highly influential in public-policy circles [1], and was recently incorporated into Google's services [43].

While the legal issues surrounding GDELT are opaque (the request of one author for clar-

ification was met with an ambiguous response of "it's a touchy subject"), it appears that one of the main developers of GDELT may have utilized copyrighted resources purchased by the University of Illinois' Cline Center for developing the SPEED dataset. It should be noted that we only utilize the aggregate historical data for comparing GDELT with other event data projects, and only utilize the publicly available articles for our validity analysis. In no way have we accessed any copyrighted material purchased by either the Cline Center or GDELT. As far as we are aware there is no ongoing litigation surrounding the use of GDELT and the data has since been re-established on the web. It has also been utilized by government agencies, Google, and continues to produce reports for Foreign Policy and other publications [66, 65, 1, 43].

Nonetheless, the GDELT controversy illustrates the importance of having a corpus that can be utilized freely by event data teams. A collective effort, organized by a government agency or consortium, to collect such a corpus would help to avoid the use restriction issues that prevent all but the very best funded teams from working on event data projects and would avoid the legal ambiguity that gave rise to the GDELT controversy. Unlike many text analysis projects, such as analyzing the collected works of Shakespeare, copyright issues loom large in analysis of news articles. The GDELT case demonstrates just how difficult navigating some of these issues can be.

3.1.3 More Recent Advances

As outlined in [103, 26], there is now an additional event data program under way - the Open Event Data Alliance (OEDA). The OEDA is attempting to produce a more transparent and open event data format than ICEWS, without the baggage that came with GDELT. It functions as a professional organization to promote open and transparent event data collection. There are currently four pieces of software being produced by the OEDA [26]. EL:DIABLO is designed as a modular system for acquiring web-based news sources and coding them into event data. PETRARCH is a program, originally started around the same time as the launch of GDELT, which is an open source Python-based event coder. Finally, the OEDA has developed a scraper and pipeline to collect news information and move the data through the process.

This effort is fantastic and need to be supported, but it is insufficient for producing data on the scale of that produced by ICEWS and GDELT. While the increasingly online nature of news media is quickly allowing more and more access to the corpus resources needed to compile event data, large-scale efforts will still need support in dealing with copyright issues and massive downloading. Moreover, OEDA's efforts still have not reached the scale necessary to overcome many of the known issues. Finally, there is the question of timeframe. One of the most exciting aspects of ICEWS and GDELT was their temporal coverage (from 1940 in the case of GDELT and from 1991, although only since 1995 is available publicly, in the case of ICEWS). These resources are not readily available online as of yet.

3.1.4 Should Event Data Processing be Fully Automated?

One of the outstanding issues in this field is to what extent event data coding should be automated. Some have argued that a semi-automated approach, which combines computer-generated codes with human supervision will produce more accurate results [44, 108]. This is the basis, for example, of the framework that generates SPEED.

The two options are clearly not mutually exclusive. One of the reasons why GDELT and OEDA have emphasized openness of their corpus (at least when hyperlinks are available) is because of the ability for others to evaluate their coding decisions and recommend modifications. One could even imagine a system where a dedicated group of analysts are continually able to modify the coding of events to make continuous improvements to the learning algorithm underlying event data generation.

Producing a less autonomous system may improve accuracy, but it does come at a cost. First, the speed and scope of events coded would vary proportionately to the level of automation. Having individuals examine and re-code a large number of event reports is likely to be costly and potentially slow. Second, to the extent that the analysts are analyzing a large number of events with a large number of coding options, the results could actually be less accurate than machine coding. This is especially true in the "normal" circumstances of human event coding, where relatively lightly trained (and poorly paid) undergraduates attempt to code across the entire CAMEO coding scheme [28]. As we note in the discussion of the GSR below, using professional coders across a limited range of events is an expensive undertaking.

3.2 Reliability Experiments

3.2.1 Event Encoding Dataset

The Integrated Crisis Early Warning System (ICEWS) is a proprietary system and was intended to forecast international crises for US analysts. It was created by Defense Advanced Research Projects Agency (DARPA) in 2007, and has been funded by Office of Naval Research (OCR) since 2013. The ICEWS project is currently maintained by Lockheed Martin Advanced Technology Laboratories. It is currently available on the Harvard Dataverse with a 1 year delay in posting new events.

The Global Database of Events, Language, and Tone (GDELT) was created by Kalev Leetaru, Philip Schrodt and others. The entire GDELT database is fully free to access. Currently, it includes approximately 270 million events since 1979. Since April 1, 2013, GDELT provides daily updated event data files with updates posted at 6 AM EST, 7 days a week.

The Social, Political and Economic Event Database (SPEED) from the University of Illinois' Cline Center for Democracy uses a combination of human and automated coding, along with an expansive Global News Archive. The hybrid system starts with humans coding thousands of news articles. These are then used to formulate a statistical model for whether an article identifies a particular type of event. This model is then utilized to analyze millions of other articles from the Global News Archive. Articles which fall above the statistical threshold are kept and those below the threshold are discarded. The internal evaluation of these systems suggests a false negative rate of 1- 3%, depending on the source. Further information can be found in the SPEED whitepapers [44, 37, 38].

The Gold Standard Report (GSR) dataset is provided by MITRE corporation. It was developed as a ground truth dataset for the Intelligence Advanced Research Projects Activity's (IARPA) Open Source Indicators (OSI) program. The GSR focuses on protest events covering 10 Latin American countries, and includes event description, location and timestamp of the first mention by major news source. All the events are manually encoded by professional MITRE analysts (not students) from a set of English and Spanish-language news sources. The development of the GSR is also an indication of the need for a coordinated policy effort to improve event data. For most event coding studies, the baseline comparison

is against a small team of undergraduate coders who are coding across the entire CAMEO coding scheme [60]. This is because almost all programs involving event data coding cannot afford to develop a ground truth dataset like the GSR, with professional coders focused on a particular set of events.

3.2.2 Correlation Between ICEWS, GDELT, and GSR

In this section, we study the correlation between GDELT, ICEWS, versus hand-coded protest sets (GSR). The GSR dataset is generated by a team in MITRE, who hand code the news report from local and international newswires for protest events in Latin America since 2011. The GSR dataset covers 10 Latin American countries: Argentina, Brazil, Chile, Colombia, El Salvador, Ecuador, Mexico, Paraguay, Uruguay and Venezuela.

We start with the correlation between GDELT and GSR. As shown from Table 3.1, the level of correlation depends heavily on the country being investigated and the level of aggregation. Countries with a larger degree of Western media coverage (e.g. Argentina, Brazil, and Venezuela) have a higher level of correlation between GDELT and GSR, which should not be surprising, given the reliance on English- language news sources in GDELT. The correlation also improves if we look at the monthly versus the weekly or daily correlations.

Figure 3.1 illustrates these correlations for all countries. The y-axis is the GDELT weekly count of events and the x-axis is the GSR weekly count. This figure illustrates some of the central problems in the cases where GDELT disagrees substantially with the GSR. First, the English language corpus results in fewer events being detected in places where there is not a lot of Western interest. Second, there are severe outliers, which are likely the result of the duplication process, which magnifies the errors.

We also study the correlation between ICEWS and GDELT. The results shows relatively higher correlation than the correlation between ICEWS and GDELT than between GDELT and the GSR. The reason may be that both ICEWS and GDELT primarily use international English newswires as their sources and operate on relatively similar coding frameworks. The ICEWS and GDELT correlation is shown in Table 3.2.

Finally, Table 3.3 reports the correlation between ICEWS and GSR events. The results are very similar to the comparison with GDELT. This is not surprising, given that they use

Table 3.1: Correlation between GDELT and the GSR

GDELT VS GSR Correlation			
Country	**Daily Count Corr**	**Weekly Count Corr**	**Monthly Count Corr**
Argentina	0.0409 (0.0366)	0.3200 (0.0284)	0.4638 (0.1480)
Brazil	0.2012 (0.1574)	0.4146 (0.1498)	0.6061 (0.4583)
Chile	0.0196 (0.0053)	0.0986 (0.0365)	0.3445 (0.3835)
Colombia	0.0126 (0.0013)	0.3034 (0.0652)	0.3906 (0.2816)
Ecuador	0.0001 (0.0001)	0.0042 (0.0001)	0.0543 (0.0423)
El Salvador	0.0070 (0.0001)	0.0021 (0.0120)	0.0736 (0.0619)
Mexico	0.0776 (0.0722)	0.2998 (0.2457)	0.4315 (0.3361)
Paraguay	0.0001 (0.0001)	0.0019 (0.0001)	0.0481 (0.0324)
Uruguay	0.0003 (0.0006)	0.0157 (0.0532)	0.1486 (0.0311)
Venezuela	0.4523 (0.2406)	0.7583 (0.5201)	0.8539 (0.6589)

Table 3.2: Correlation Between ICEWS and GDELT

GDELT VS ICEWS Correlation

Country	**Daily Count Corr**	**Weekly Count Corr**	**Monthly Count Corr**
Argentina	0.0424	0.0841	0.1393
Brazil	0.3561	0.5706	0.6769
Chile	0.237	0.4057	0.2204
Colombia	0.3034	0.6333	0.6041
Ecuador	0.0066	0.006	0.0414
El Salvador	0.0738	0.3919	0.6136
Mexico	0.0766	0.2457	0.4189
Paraguay	0.0012	0.0007	0.003
Uruguay	0.0046	0.0047	0.1165
Venezuela	0.6813	0.827	0.9613

ostensibly similar coding frameworks and primarily English-language corpuses.

Figure 3.2 shows these correlations on the weekly level graphically. Again, the results highlight the difficulty ICEWS has picking up events in states without significant Western re-

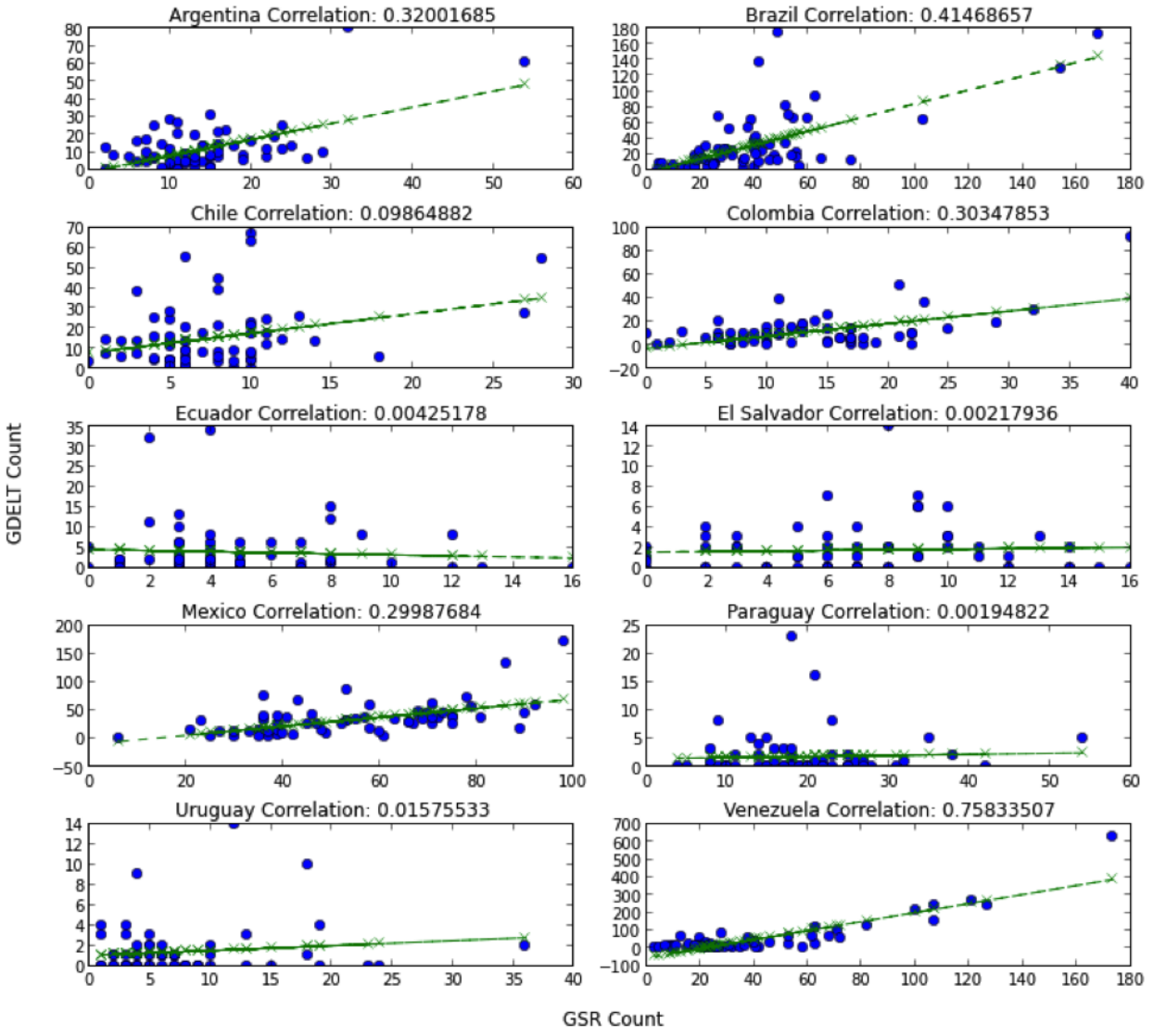

Figure 3.1: Correlation plot between GDELT and GSR events

porting and the magnification of errors due to duplication.

In sum, these results suggest major discrepancies between automated systems that use a primarily English-language corpus and a hand-coded system that uses news sources in both English and Spanish. This emphasizes several of the points made in the main article. While these data efforts are heroic, there are some predictable blind spots that need to be addressed in a systematic way in future event data efforts.

Table 3.3: Correlation Between ICEWS and GSR

ICEWS VS GSR Correlation

Country	Daily Count Corr	Weekly Count Corr	Monthly Count Corr
Argentina	0.0135	0.0453	0.0004
Brazil	0.2827	0.7242	0.8416
Chile	0.0053	0.0538	0.0763
Colombia	0.0375	0.3403	0.2374
Ecuador	0.0207	0.0861	0.3133
El Salvador	0.0003	0.0053	0.0641
Mexico	0.0051	0.0629	0.0044
Paraguay	0.0004	0.0004	0.0104
Uruguay	0.0005	0.2210	0.4625
Venezuela	0.4101	0.7534	0.8601

3.2.3 Correlation between ICEWS, GDELT and SPEED

This analysis compares the SPEED dataset - a hybrid human-coded and automated dataset of civil unrest from 1946 to 2005 - against ICEWS and GDELT. The data for both of these were processed so that they would only cover the same countries and the same years: 1991-2005 for ICEWS and 1979-2005 for GDELT. The unit of analysis is country-day.

The key element of this analysis is that we are not going to be looking at raw correlation (for which results are equally bad), but rather whether the datasets report any event on the day that another dataset reports a protest. This is because SPEED focuses on whether an event matching its criteria occurs on a particular day. We would expect for these datasets to be rather similar in recording whether any event happens on a particular day. We might suspect that ICEWS and GDELT would have more false positives. As shown below, however, there is a surprising absence of such a pattern in the differences between the datasets.

To start, we compare whether the datasets recorded an incident of civil unrest in a particular day. Table 3.4 looks at a comparison of when GDELT records a civil unrest event versus when SPEED records a civil unrest event. We do not record true negatives, since this would encompass all days for which neither records a civil unrest event for any country (by

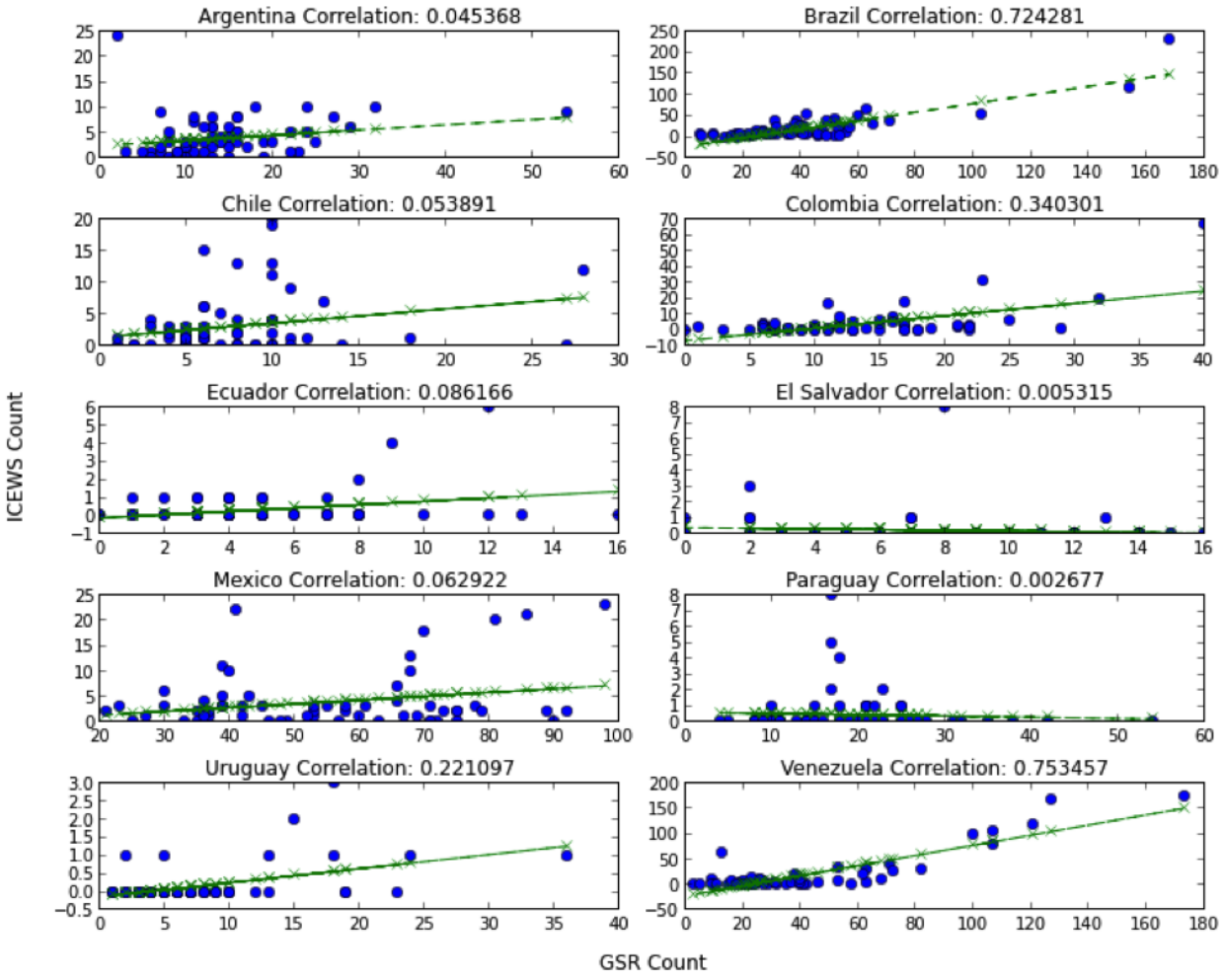

Figure 3.2: Correlation plot between ICEWS and GSR events

definition, a large number and one that is relatively uninteresting to policy-makers).

Table 3.4: Comparison of SPEED with Raw GDELT in Indicating and Event

	GDELT		
SPEED		Event Recorded	Event Not Recorded
	Event Recorded	47989 (17.2%)	87323 (31.3 %)
	Event Not Recorded	143578 (51.5%)	

GDELT also provides a filter to indicate whether a recorded event is a âĂIJroot event.âĂİ This is designed to only capture events that are particularly important in news reporting

and is meant to help address the false positive problem. Table 3.5 does the same comparison only looking at those events labeled as root events.

Table 3.5: Comparison of SPEED with GDELT âĂIJRoot EventâĂİ Indicator

	GDELT		
SPEED		Event Recorded	Event Not Recorded
	Event Recorded	36793 (15.2%)	98519 (40.7 %)
	Event Not Recorded	106942 (44.1%)	

Finally, it is possible that the number of reports in GDELT gives some indication of the issues involved. One interpretation of GDELT counts is that the more recorded events in a particular day, the more important that event was. Perhaps SPEED, because of the human coding aspect, only records the events that are very important. If so, there should be a positive correlation between the count of events in GDELT and the recording of an event in SPEED. In fact, the opposite is true. There is a correlation of - 0.124 for the raw GDELT counts and -0.146 for the root event counts. This is substantially better than the correlation just using the indicator variables, meaning that higher numbers of reports is a somewhat better predictor of whether SPEED records an event than raw counts, but the negative correlation is still troubling.

ICEWS already applies several filters to avoid false positives. Table 3.6 shows the results comparing ICEWS to SPEED in recording events.

Table 3.6: Comparison of SPEED with ICEWS in Indicating and Event

	ICEWS		
SPEED		Event Recorded	Event Not Recorded
	Event Recorded	8764 (10.3%)	35704 (31.9 %)
	Event Not Recorded	40745 (47.8%)	

Again, thinking that perhaps the number of recorded events, indicating the salience of the event, might give us a good predictor of when SPEED would record an event, we looked at the correlation. In this case, it was -0.257.

The SPEED dataset is primarily constructed for accuracy and validation, it is therefore quite

conservative in what it records as a civil unrest event. With that being said, these results are troubling; even-more-so for the large numbers of false negatives in both ICEWS and GDELT – days in which SPEED recorded a significant civil unrest event, but the computer coded sources did not. Considering that SPEED and GDELT use similar sources, this is problematic and suggests that the problems may be more than in yielding false positives.

3.2.4 An Analysis of GDELT Sources

In addition to relying on an English-language corpus, we also noted the GDELT is significantly reliant on a few sources within this corpus. Our analysis of the news sources listed by GDELT shows that a small number of domains contribute most of the events in GDELT. As shown in Figure 3.3, the first 10% domains contribute more than 80% of the events in our experiment set. As shown in Figure 3.4, the events to domains distribution follows a power law distribution. Most of the domains generate very few events, but some sites generate a huge number of events.

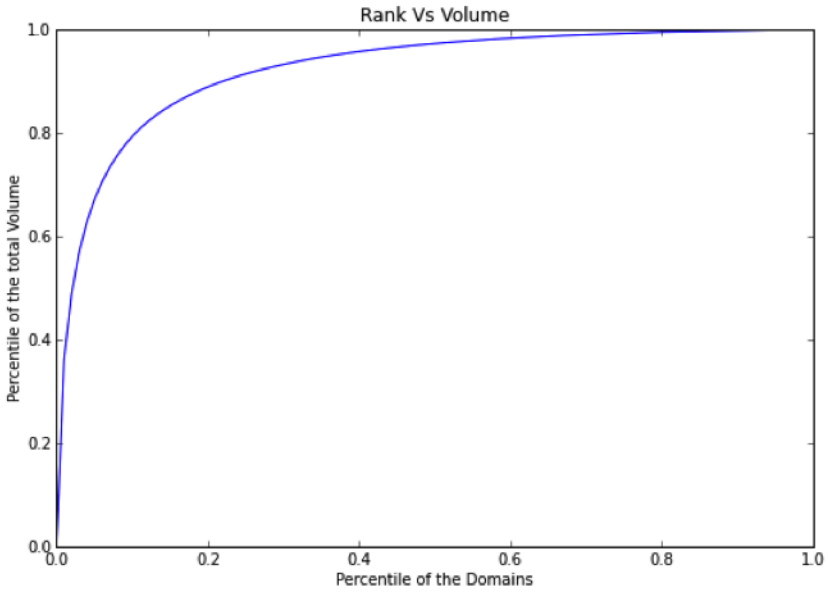

Figure 3.3: GDELT sources Rank Vs Volume

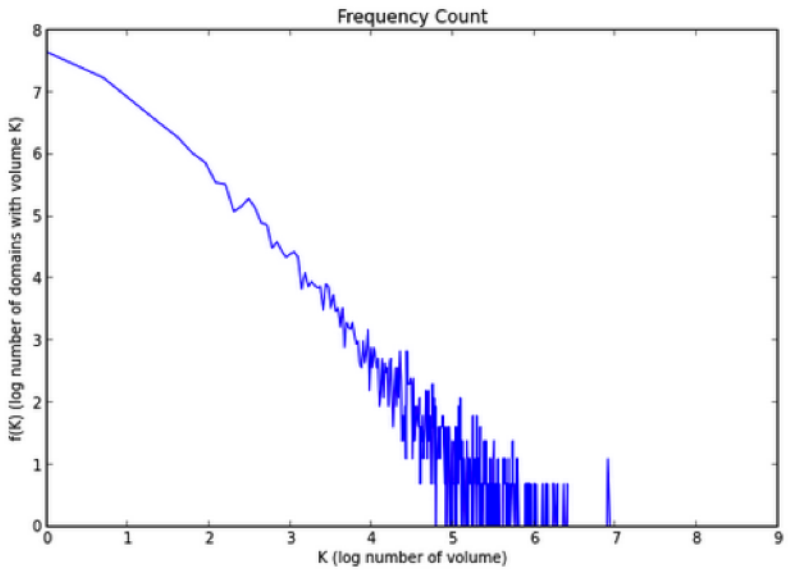

Figure 3.4: GDELT sources Volume Vs Domains

3.3 Validility Experiments

3.3.1 GDELT Data Set

We download the GDELT daily files starting from 2013-04-01 to 2014-06-02 and extract all the events with EventRootCode 14 (protest). We obtain a total of 431,539 records. Among these records, we remove those events with invalid SOURCEURL not starting with http or https (eg. BBC Monitoring, MENA news agency, Cairo/BBC Monitoring) and obtain a total of 416,336 records. This may be caused by restrictions due to the news copyright. For each remaining event, we are trying to access the source url and extract the author, title, summary, lead paragraph and full text. During the content extracting process, we find that a lot of URLs are not valid anymore and return the 404 (Page not founded) HttpErrors. Figure 3.5 show the summary of the protests records over URL. Finally, we obtain 344,481 records with proper content extracted from the source url.

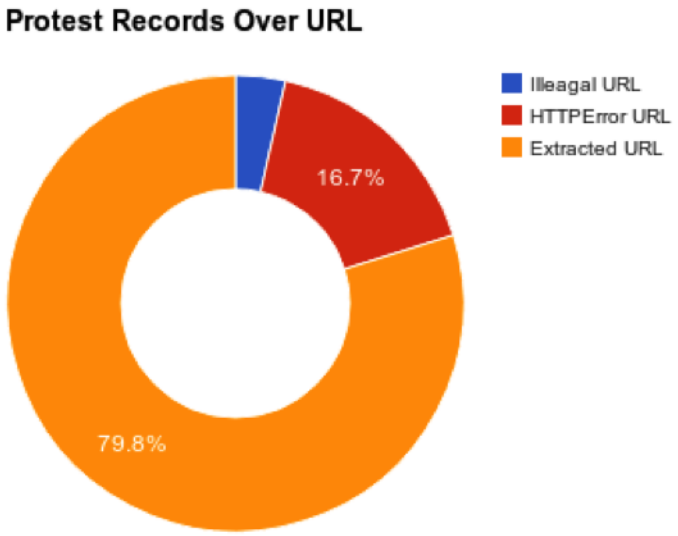

Figure 3.5: GDELT Protest records distribution over type of URLs

3.3.2 Data Clean Up

We use a list of 55 protest keywords designed by the domain experts for keyword filtering. The keywords we use are listed in Figure 3.6

Protest Keywords						
demonstrators	civil disobedience	parade	demonstrations	disorder	rally	barricade
civil unrest	mob	parades	protests	strike	rallies	barricades
civil disturbance	mobs	march	protested	strikes	sit-in	roadblock
civil disturbances	protesters	marches	protesting	striking	sit-ins	roadblocks
civil disruption	tear gas	marched	unrest	strife	clashed	demonstrate
civil disruptions	demonstrators	marching	disturbance	blockade	clashes	demonstrated
civil disorder	strikers	protest	disruption	blockades	attacks	demonstrating
civil strife	marchers	demonstratio	disrupted	blockaded	conflict	

Figure 3.6: Domain experts predefined protest keywords

We combine the title, summary and the content as the full text of the event and divide them into sentences. If none of the sentences of the event contain any one of the protest keywords, then we label the event as false positive and remove it from the candidate event set. After

the keywords filtering step, we obtain 253,720 (73.7%) out of 344,481 records in which the full text of the event contains at least one of the protest keywords.

After keyword filtering, we utilize the Stanford Temporal Tagger (SUTime) to perform the temporal entity filtering. For each sentence containing the protest keywords, we extract the neighbor sentences and use SUTime to extract the temporal entities from them. If the sentence doesn't have any temporal entities surrounding the event, we infer that the event sentences only mention the protest topic and label it as a false positive. Once we get the temporal entity, we can compute the relative date according to the date of the article's post. Moreover, we are more interested in the recent events, so we only keep those sentences with temporal entity within the one month time window since the article posted. If none of the event sentence satisfy the temporal filtering, we would label the event as false positive. We obtain 178,987 out of 253,720 records which pass both the keywords and temporal entity filtering stages.

3.3.3 Event Deduplication

Previous research on the GDELT event set have demonstrated a serious problem of event duplication. Especially when the same event is reported by more than one media source, GDELT often encodes them as multiple events. For example, there was a post in FiveThirtyEight trying to utilize the GDELT event set to analyze kidnapping trends in Nigeria. One problem of the report is that it interpreted the media reports of kidnapping as the "number of kidnappings" due to the event duplications in GDELT set. In this case, the duplicates would inevitably give an inflated estimate of the real events when there are one or more high profile incidents. The situation becomes even worse as events grab more media attention. Even when more than one event is derived from a single url, the duplications still exist in some cases. Moreover, in some cases, the consecutive reports for the same event will also be encoded as multiple events.

In this work, we propose a method to conduct the event deduplication based on the similarity between events. We consider two events are duplicated if their similarity is higher than a specific threshold. We use eventDate, ActorCode, ActionType, ActionGeoFullName and event sentences as features for each event. We set up a time window w, and for all the events

in that time window, we firstly group the events by the locations. We then compute the similarity between each pair of events in the location group and utilize a greedy strategy to remove the duplicated events. For this study, we set the similarity threshold as 0.8 and we obtain 113,932 out of 178,987 unique events.

3.3.4 Protest Event Classification

Among these remaining events filtered by the previous steps, we randomly choose 1000 event sentences for manual inspection. We found that the sentences could be classified into three categories: protest, non-protest and planned protest. The proportion of each category is shown in Figure 3.7. Examples of each of these types of events are shown in Table 3.7.

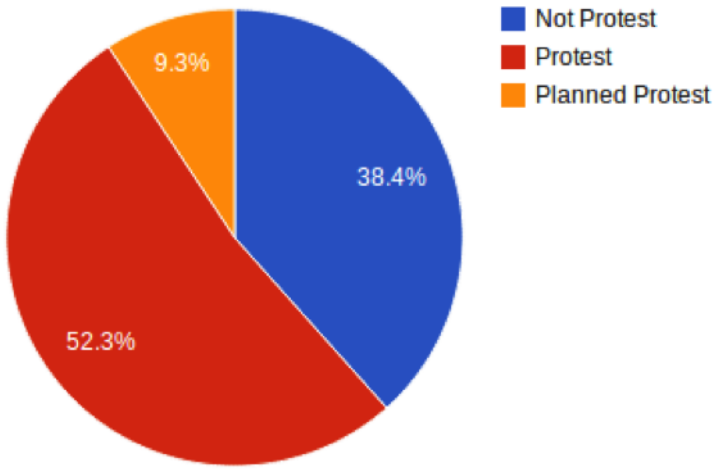

Figure 3.7: proportion of three types of sentences in training set

We use the Stanford Dependency parser to extract the collapsed typed dependency tree for each event sentence. These dependencies provide a tree structure to represent the grammatical relations between words in a sentence. They comprise a set of triplets: relation name, governor and dependent. Specifically, we use the collapsed typed dependencies tree to represent the sentence structure. A typical example of the collapsed typed dependencies tree of the sentence is shown in Figure 3.8.

Since all the sentences contain the protest keywords in this stage, the words surrounding the keywords would take an important role to distinguish the sentences among these three

Table 3.7: Examples of events categorized as protests, non-protests, and planned protests.

Event Category	**Sentences**
protest	Hundreds of bank workers protested outside parliament on Thursday, worried that they could lose much of their pension savings under the terms of the bailout deal.
protest	Several protesters were wounded in the clashes on Wednesday, the first in about three months in the oil-rich Gulf state which underwent violent protests late last year against the amendment of an electoral law, witnesses said.
non-protest	The police had on Wednesday contacted Mr Goh regarding his Facebook post calling on the public to deface a poster of Prime Minister Lee Hsien Loong at the demonstration.
non-protest	On Saturday, it said further protests would not be allowed.
planned protest	A new protest has been scheduled for Tuesday at the Zlatograd border crossing point on the border with Greece.
planned Protest	In a further sign of tension, Istanbul's governor said Sunday that a planned demonstration in Taksim Square would not be allowed to go ahead.

categories. For each sentence, we extract the k-depth neighbor words of the matched protest keywords as candidate features. However, we are not interested in all possible types of relation in the Stanford dependencies. We only use the edges with grammatical relations like obj, dobj, iobj, pobj, subj, nsubj, nsubjpass, csubj, csubjpass, amod, advmod, neg, aux and nn. Moreover, we do not perform any stemming or lemmatization to the words, since the tense of the words may indicate the type of the events.

Once we extract the features from training set, we use a SVM classifier to classify the remaining event sentences into three categories: protest, non-protest and planned protest. Joachims [53] has discussed the advantages of using SVM for text classification:

- SVM provide overfitting protection and would be able to handle a large number of features.

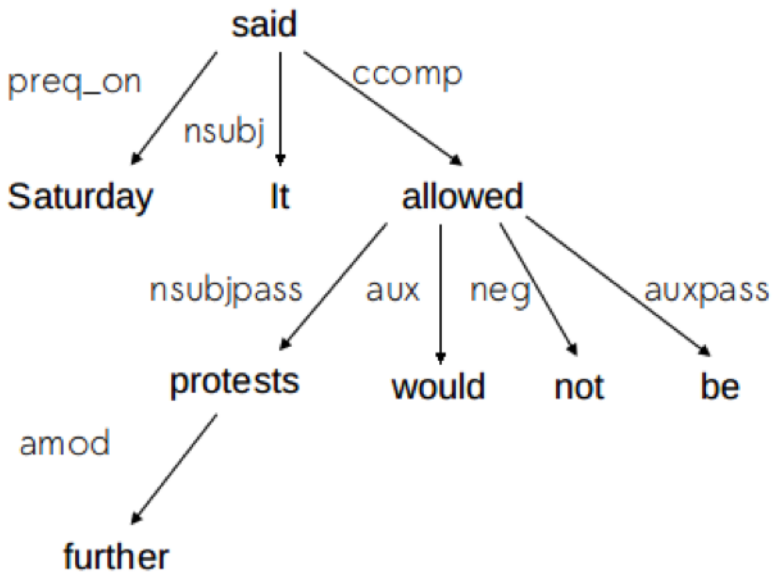

Figure 3.8: Example of dependency parse tree

- In text classification, there are very few irrelevant features and it is preferable to begin with as many features as possible.
- The document vectors are sparse and SVM is well suited for problems with sparse instances.
- Most text classification problems are linearly separable.

In our case one event may involve multiple sentences, so we would label the event as protest if any one of the sentences is classified as protest. We compared the use neighbors in the dependency tree as features with two other methods: using all the words in the sentence and using neighboring word based on their location in the sentence. We utilize a linear kernel SVM and conduct 3-fold cross validation on the manually labeled sentences. We obtain the best performance when using the neighbor words in the collapsed typed dependency tree as feature. The result is shown in figure 3.9.

Finally, we apply the SVM model to the remaining event set and obtain 72,210 protest events. We list event reduction of each steps in Table 3.8. The number of percentage in the parenthesis is the ratio of number of records after each operation to the original size of data set (431539).

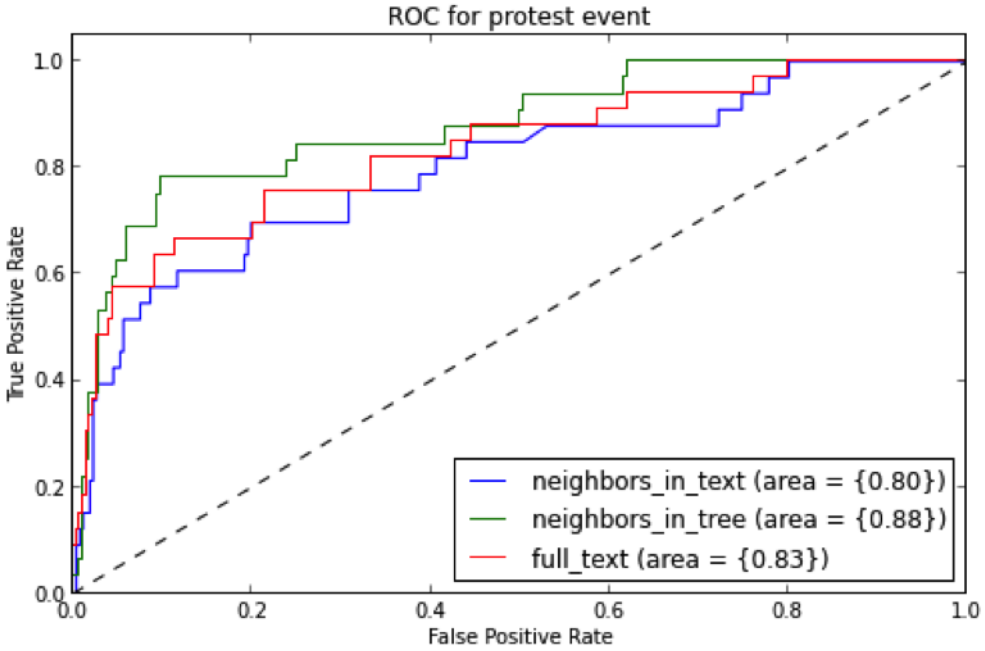

Figure 3.9: ROC for classification result on manually labeled set

Table 3.8: GDELT full event processing results

Steps	**Number of Records Before Operation**	**Number of Records After Operation**
Url Validation	431539	344481 (79.8%)
Keyword Filtering	344481	253720 (58.8%)
Temporal Entity Filtering	253720	178987 (41.5%)
Event Deduplication	178987	113932 (26.4%)
Event Classification	113932	72210 (16.7%)

Intuitively the most important event would appear in the beginning part of the news. The GDELT system uses the isRootEvent flag to indicate whether the event is extracted from the lead paragraph of the document. We perform the same sequence of experiments on these Root Events and obtain the results shown in Table 3.9.

The results show that only 16.7% for all events and 15.7% for Root Events are finally labeled as positive protest events. These results give us insight into the noisiness of GDELT for protest event encoding.

Table 3.9: GDELT root event processing results

Steps	**Number of Records Before Operation**	**Number of Records After Operation**
Url Validation	286475	225809
Keyword Filtering	225809	173442
Temporal Entity Filtering	173442	132889
Event Deduplication	132889	65836
Event Classification	65836	44937 (15.7%)

3.3.5 Experiments on ICEWS Protest Events

The ICEWS data set provides the sentence from which the event was encoded. So we perform a similar data cleaning experiment to assess ICEWS events. Since we can't access the full content of the news article from which ICEWS events are being extracted, we are unable to verify the portion of the article from which the event date is being determined. Therefore we only apply the keywords filtering, event classification and deduplication stages on the ICEWS events set. Table 3.10 shows the experiment results.

Table 3.10: ICEWS event processing results

Steps	**Number of Records Before Operation**	**Number of Records After Operation**
Keyword Filtering	58489	48574
Event Deduplication	48574	39186
Event Classification	39186	31718

The classification result shows that around 80% of the keywords filtered events are classified as protest events, which implies that the ICEWS encoding system has significantly high precision over GDELT. There are around 20% duplicated events in ICEWS set, one common reason being that different websites often report the same event(s) with different usage of expressions.

3.3.6 Correlations After Filtering

A natural question to ask is whether the event data from GDELT and ICEWS have a closer correlation with the GSR after they have gone through the filtering process described above. Interestingly, we do not find a consistent improvement. As the reader will note in Table 3.11, the correlation, based on weekly event counts, improves in some cases and worsens in others. The average correlation is actually somewhat lower after filtering (although it is wholly dependent on the result in Colombia, and reverses if this outlier is excluded). While it is frustrating to note that we cannot solve GDELT's issues with some further processing, it is not completely surprising, given, as we note above, that much of the problem lies in the corpus used to create GDELT.

Table 3.11: Comparison of correlation between GDELT and GSR before and after filtering.

Country	**Before Clean**	**After Clean**
Argentina	0.32	0.4041
Brazil	0.4146	0.3672
Chile	0.0986	0.1257
Colombia	0.3034	0.0713
Ecuador	0.0042	0.0651
El Salvador	0.0021	0.0054
Mexico	0.2998	0.2174
Paraguay	0.0019	0.004
Uruguay	0.01575	0.031
Venezuela	0.7583	0.7667

3.4 Errant Cases Analysis

Table 3.11 shows errant cases using the "root events" listed by GDELT. These are the events listed in lead paragraphs and are considered the most likely to actually represent a true event. In the Table, we list the error type, the event ID, the lead paragraph, a comment on the error and the (at the time of this writing) valid URL for the article.

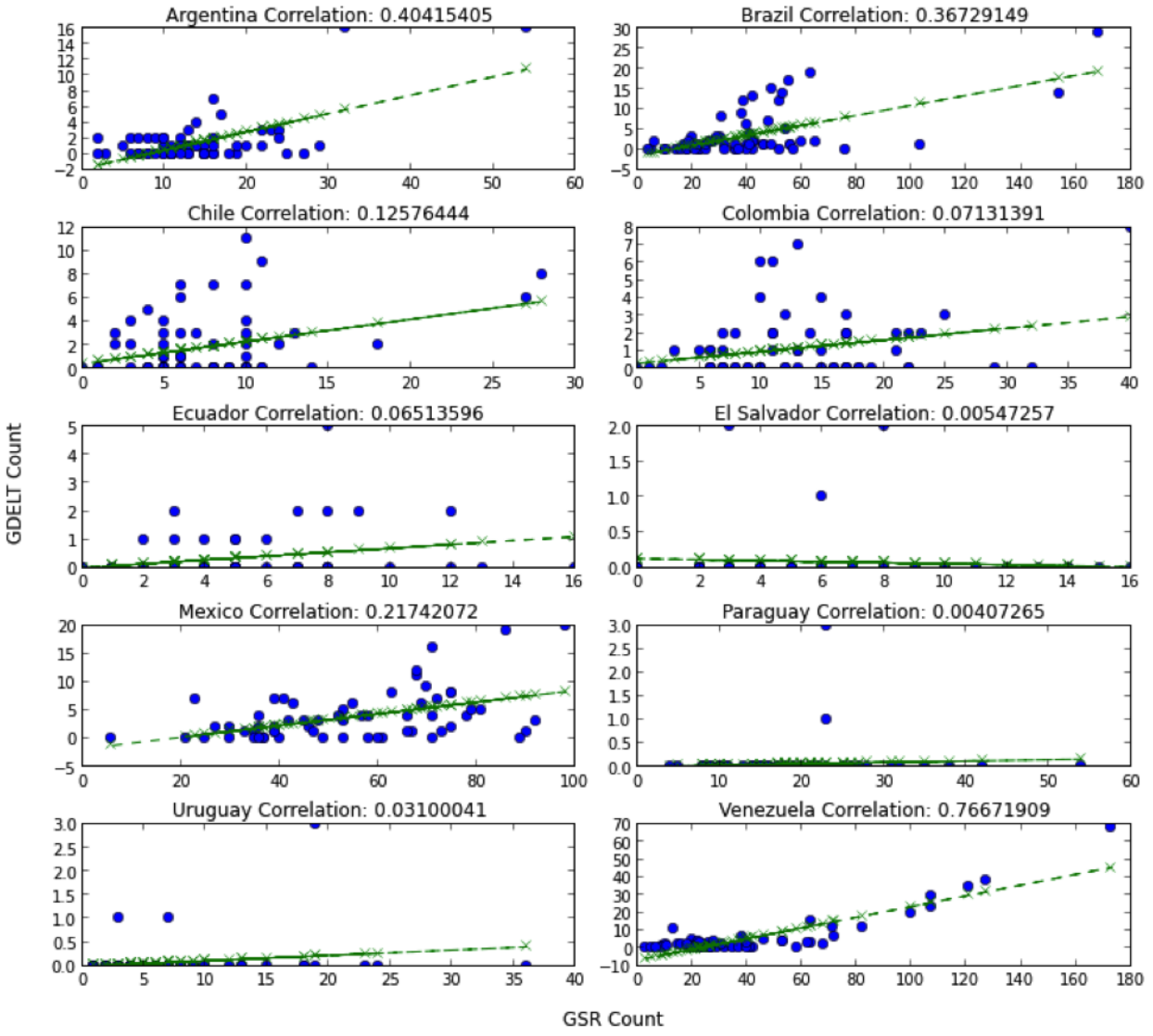

Figure 3.10: Correlation matrix between filtered GDELT data and GSR data in weekly counts.

Table 3.12 repeats this exercise for ICEWS. Since ICEWS only reports the event sentence, this is included instead of the lead paragraph and URL. Again we provide the error type, event ID, event text, the sentence from which it is drawn, and some comments on the error.

Error Type	Lead Paragraph	Error comments
Location Error	WASHINGTON — The White House on Wednesday denied that a staff member's email three days after the deadly attack on the U.S. mission at Benghazi, Libya, was actually about the attack. Critics have branded the electronic missive as evidence that the Obama administration sought to deceive the public about the true circumstances surrounding the deaths of four Americans during the final months of the 2012 presidential campaign.	The GDELT event has been geolocated in Cario, Al Qahirah, Egypt. But according to the content of the post, the news has nothing to do with Egypt.
Polysemy Error	Pope Francis is hoping to demonstrate the power of prayer next week when Israeli President Shimon Peres and Palestinian President Mahmoud Abbas join the pontiff at the Vatican for an exercise in peace building.	The GDELT event was mis-encoded as protest event may due to the misunderstanding of the keyword "demonstrate".
Planned Protest	After word spread about an environmental protest that was planned for Saturday in the central Chinese city of Chengdu, drugstores and printing shops were ordered to report anyone making certain purchases. Microbloggers say government fliers urged people not to demonstrate, and schools were told to stay open to keep students on campus.	The GDELT event was mis-encoded may due to ignore the word "planned"
Cancelled Protest	A Ukrainian court has blocked the country's first gay pride demonstration in the centre of the capital, upholding a complaint by authorities that the rally would disturb annual Kiev Day celebrations and could spark violence.	The GDELT event was mis-encoded may due to ignore the word "blocked"
Without Time Information	Abdullah El-Shamy, an Al Jazaera journalist who has been in jail for nine months, has been on hunger strike for over four months to protest his detention. Local and international rights groups have called for his release.	Although the body of the news contain the keyword "protest", but it does not refer to any time information.
Unclear Reason	Chandigarh: The Punjab School Education Board (PSEB) has postponed the declaration of Class 10th exam results to Tuesday. The PSEB was earlier scheduled to declare the results on Monday. In a short note on its Website, the Board informed, "Secondary (Class 10th) Exam Result 2014 is likely to be declared on 3th June 2014 (sic)." Over 3 lakh students had appeared for the examination that was held in March-April last year.	Not clear which strategy being used to encode the article as a protest event.
Duplicated Events	UNITED NATIONS (Reuters) - U.N. Secretary-General Ban Ki-moon on Tuesday expressed alarm at the violence in Turkey as confrontations between Turkish security forces and protesters continued after three weeks of demonstrations against Prime Minister Tayyip Erdogan.	Both news site report the same event but on two consecutive days. GDELT encode them as two different events.

Figure 3.11: Examples of errant sentences in GDELT (with explanation)

Error Type	Event Text	Event Sentence	Error comments
Without Time information	Demonstrate or rally	Recently, they maintained, it has developed into a new form with students rallying for better services, but "interference by external parties" make things take a different turn.	The sentence contain the protest keywords but it doesn't indicate any specific event
Without Time information	Demonstrate for leadership change	It is a year since Egyptians, inspired by an uprising in Tunisia, took to the streets to call for reform change and to demand the resignation of Mubarak, Egypt's president for 30 years.	The sentence talks about historical protest fact but not a specific event.
Without Time information	Demonstrate or rally	The results also showed that bad economic conditions and high prices are the main factors motivating people to take part in protests, followed by feelings of injustice, inequality and concerns about safeguarding political, social and religious rights.	The sentence contain the protest keywords but it doesn't indicate any specific event
Polysemy Error	Demonstrate or rally	Since its inception, the Islamic State group has demonstrated the firmness of its structure and the strength of its organizational composition.	The ICEWS event was mis-encoded as protest event may due to the misunderstanding of the keyword "demonstrate".
Polysemy Error	Conduct hunger strike	The rationale is to launch the operation before the Muslim fasting month of Ramadan, which will start in mid-June, and searing summer temperatures, he said.	The military operation was encoded as hunger strike
Polysemy Error	"Conduct strike or boycott	This second draft law proved challenging to pass as Sunni ministers boycotted the reform.	The ICEWS event was mis-encoded as protest event may due to the misunderstanding of the keyword "boycotted"
Planned Protest	Conduct strike or boycott	"BAGHDAD (AP) \u2014 Iraq's most revered Shiite cleric is calling for unity among the country's forces battling the Islamic State group after major, Iran-backed Shiite militias pulled out of the offensive in the militant-held city of Tikrit in protest over U.S. airstrikes there.",	Call for protest
planned protest	Demonstrate or rally	Bahrain says arrests two over planned car bombings	planned car bombings
Cancelled Protest	Demonstrate or rally	Capriles has called off a march by his supporters in Caracas, saying that his rivals were plotting to \"infiltrate\" the rally to trigger violence."	The ICEWS event was mis-encoded may due to ignore the word "called off"

Figure 3.12: Examples of errant sentences in ICEWS (with explanation)

3.5 Further Analysis

Some readers may wonder if the issues found in the protest category apply to other CAMEO categories, given that terms like "protest" and "strike" are ambiguous. We do not have ground

truth event data sets and keywords terms for other type of events, so we cannot do some of the analysis that we conducted for protest events. There are, however, other ways we can analyze the other 19 types of events defined in the CAMEO codebook.

We conduct four tests to analyze the reliability and validity of the event data sets in the other CAMEO event types. First, we look at the correlation between GDELT and ICEWS, both of which code the same types of CAMEO events, generally (although ICEWS did make some modifications to how a couple of event categories are classified). While the correlations vary substantially between categories and aggregations, the correlation between the two datasets is generally poor.

Second, we compare ICEWS and GDELT to the Militarized Interstate Dispute (MIDB) dataset, which is a long-standing human coded event dataset in international relations [90]. While we do find some signal in both datasets, there are a large number of months in which major militarized interstate disputes (MIDs) occur but there are no similar events recorded in either event data set.

Third, we conduct a duplication analysis on both ICEWS and GDELT using the same method outlined above for protests. We find that there is substantial duplication across all of the event areas, not just in the protest events.

Finally, we conduct an analysis of the quality of coding in the other event code areas, with both the authors and a set of trained graduate students evaluating the coding using an automated system. We find that, while the maladies identified in the protests category affect the other 19 CAMEO categories to different degrees, they are all severely impacted by the validity issues outlined above.

3.5.1 Correlation Analysis (GDELT vs. ICEWS)

For each category, we choose the top 50 countries (by reported location of event) with most number of events and compute the daily, weekly and monthly correlation between GDELT and ICEWS for the other CAMEO categories. Table 3.13 shows the statistics of correlations over all categories. The correlations between GDELT and ICEWS are very weak for all three levels. The average of mean correlations across all categories are 0.0639, 0.1206 and 0.2223 for daily, weekly and monthly level respectively.

	Daily			Weekly			Monthly		
Event Types	**Mean**	**Median**	**STD**	**Mean**	**Median**	**STD**	**Mean**	**Median**	**STD**
MAKE PUBLIC STATEMENT	0.0909	0.0570	0.1259	0.1344	0.0661	0.1816	0.2910	0.3130	0.2163
APPEAL	0.0523	0.0163	0.0939	0.0938	0.0373	0.1450	0.2000	0.3130	0.1965
EXPRESS INTENT TO COOPERATE	0.0749	0.0441	0.0977	0.1285	0.0666	0.1687	0.2342	0.3130	0.1865
CONSULT	0.0972	0.0495	0.1257	0.1284	0.0603	0.1541	0.2243	0.3130	0.1847
ENGAGE IN DIPLOMATIC COOPERATION	0.0560	0.0328	0.0759	0.0980	0.0444	0.1323	0.2177	0.3130	0.1838
ENGAGE IN MATERIAL COOPERATION	0.0068	0.0016	0.0159	0.0282	0.0054	0.0584	0.1169	0.3130	0.1595
PROVIDE AID	0.0251	0.0051	0.0574	0.0605	0.0135	0.1172	0.1220	0.3130	0.1535
YIELD	0.0659	0.0122	0.1017	0.1222	0.0259	0.1734	0.2319	0.3130	0.2222
INVESTIGATE	0.0330	0.0084	0.0551	0.0870	0.0232	0.1428	0.1836	0.3130	0.2127
DEMAND	0.0383	0.0188	0.0608	0.0962	0.0515	0.1292	0.2585	0.3130	0.2256
DISAPPROVE	0.0729	0.0296	0.1011	0.1346	0.0385	0.1744	0.2300	0.3130	0.2223
REJECT	0.0386	0.0139	0.0713	0.0801	0.0293	0.1269	0.1541	0.3130	0.1784
THREATEN	0.0485	0.0076	0.0997	0.1300	0.0446	0.1953	0.2134	0.3130	0.2429
PROTEST	0.1207	0.0731	0.1570	0.2032	0.1066	0.2307	0.2843	0.3130	0.2726
EXHIBIT FORCE POSTURE	0.0390	0.0090	0.0662	0.1296	0.0821	0.1786	0.2383	0.3130	0.2633
REDUCE RELATIONS	0.1033	0.0349	0.1445	0.2044	0.0882	0.2315	0.2963	0.3130	0.2690
COERCE	0.0920	0.0338	0.1245	0.1508	0.0693	0.1937	0.2539	0.3130	0.2372
ASSAULT	0.0815	0.0198	0.1353	0.1282	0.0686	0.1702	0.2404	0.3130	0.2178
FIGHT	0.0982	0.0419	0.1587	0.1523	0.0802	0.2070	0.2360	0.3130	0.2363
USE UNCONVENTIONAL MASS VIOLENCE	0.0429	0.0052	0.1054	0.1214	0.0360	0.2199	0.2202	0.3130	0.2856

Figure 3.13: Correlation between GDELT and ICEWS Over 20 Categories

Figures (3.14, 3.15, 3.16, 3.17, 3.18) plot the correlations for each category across each location country. The results show that the correlation in some areas (e.g. Egypt) are reasonably consistent, but the correlations drop off significantly after the top few cases. This suggests a similar problem to what was identified above âĂŞ these event datasets agree more when there are major events in areas likely to gain coverage in English-language media.

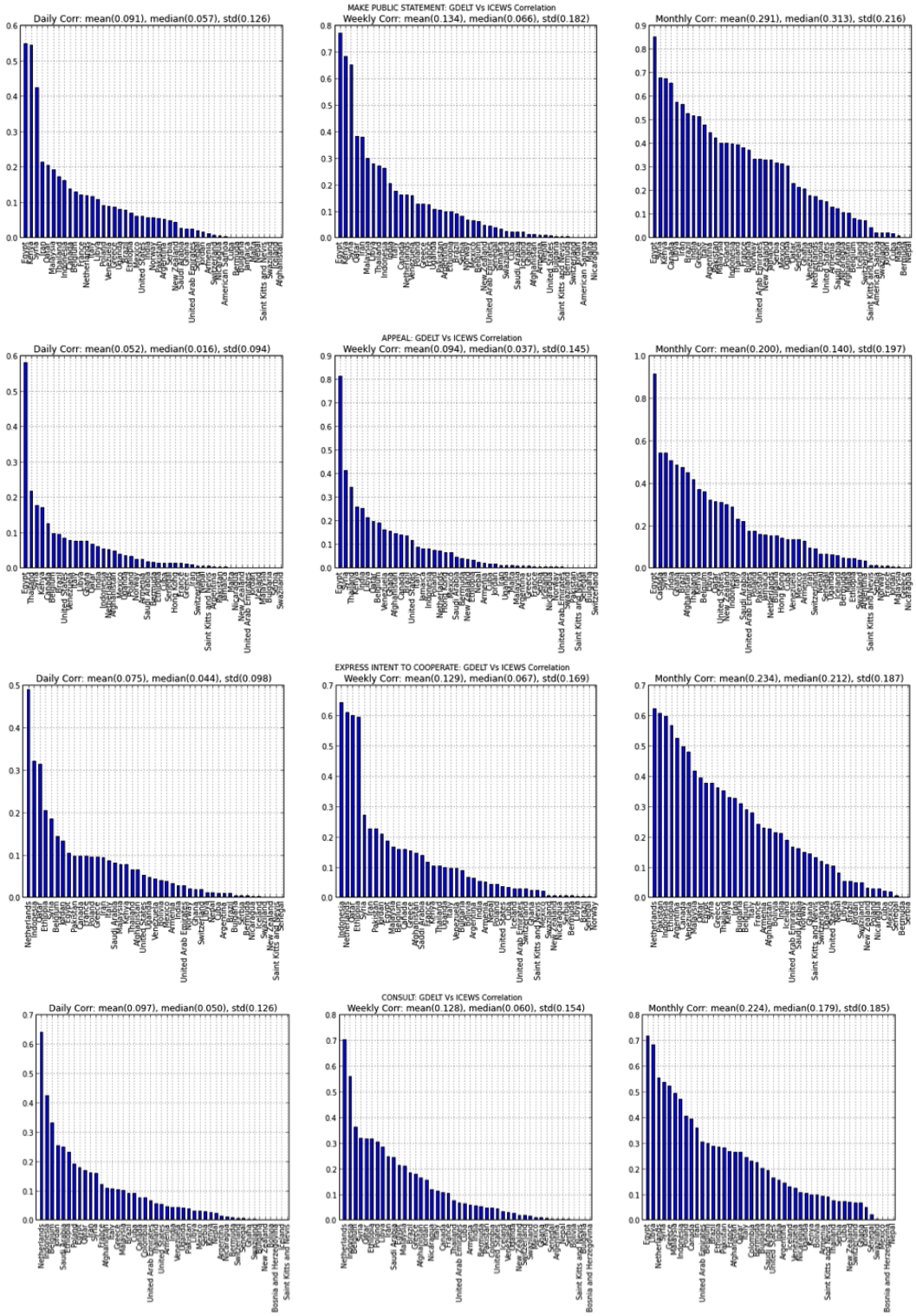

Figure 3.14: CAMEO Category 01 - 04

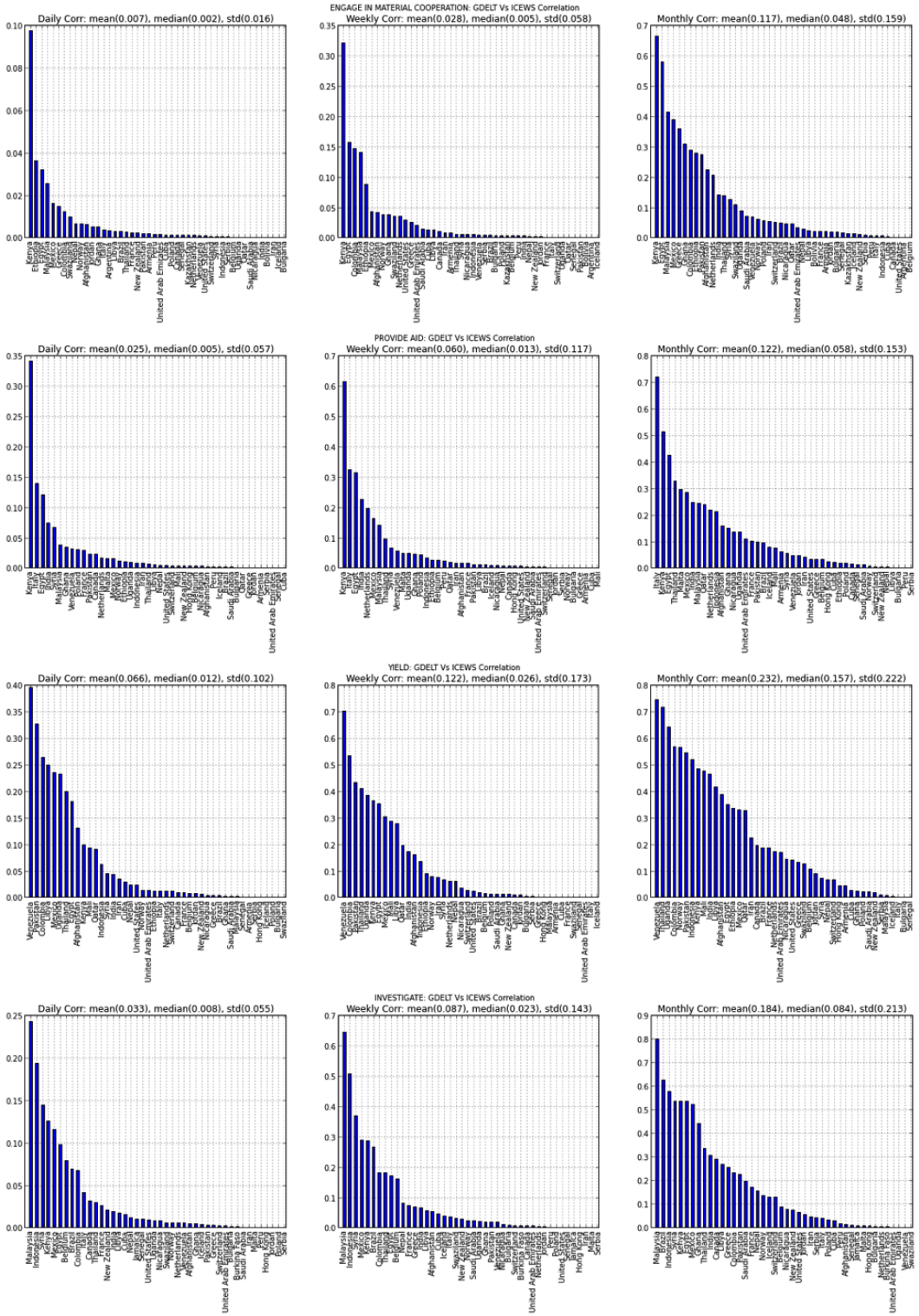

Figure 3.15: CAMEO Category 05 - 08

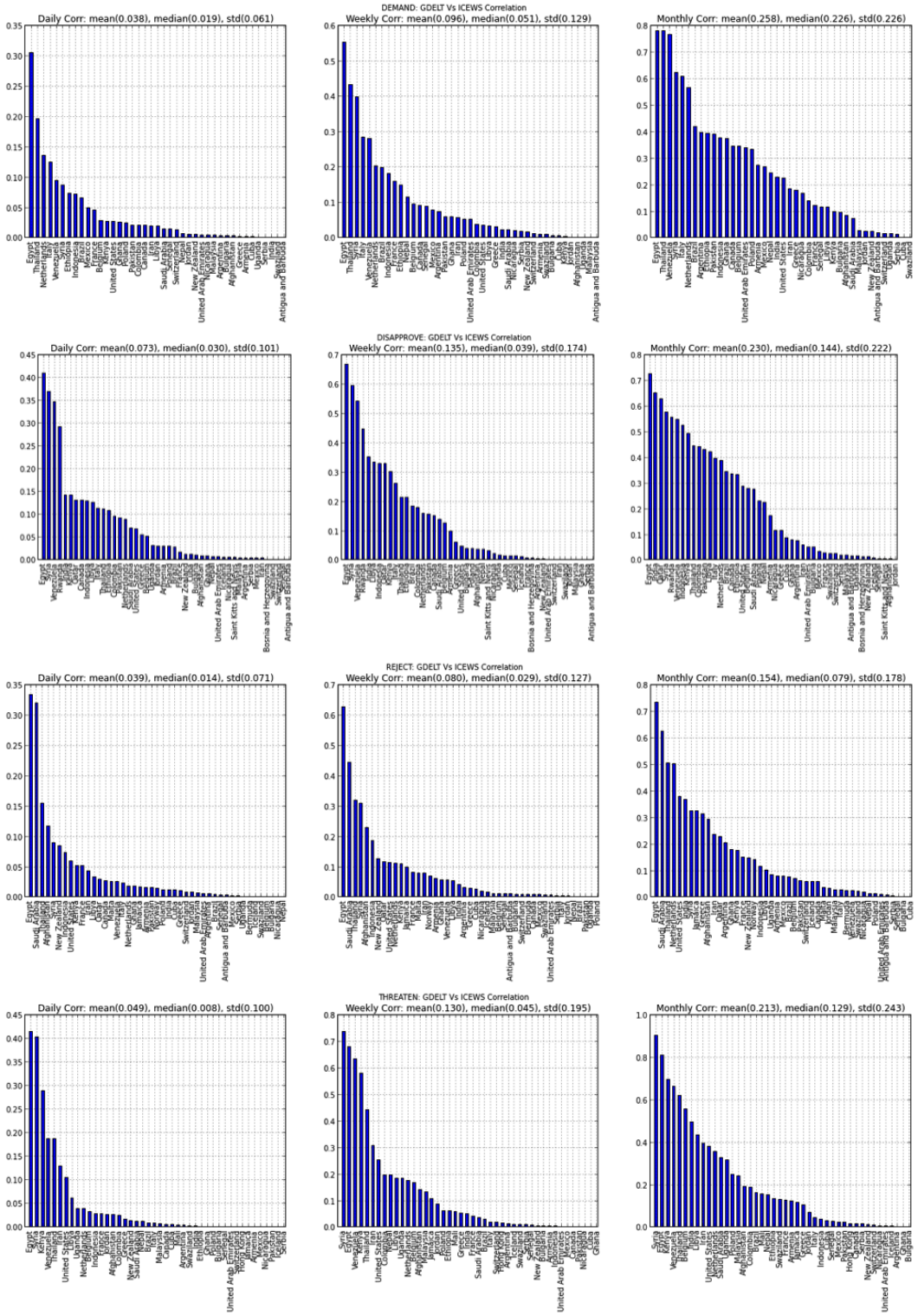

Figure 3.16: CAMEO Category 09 - 12

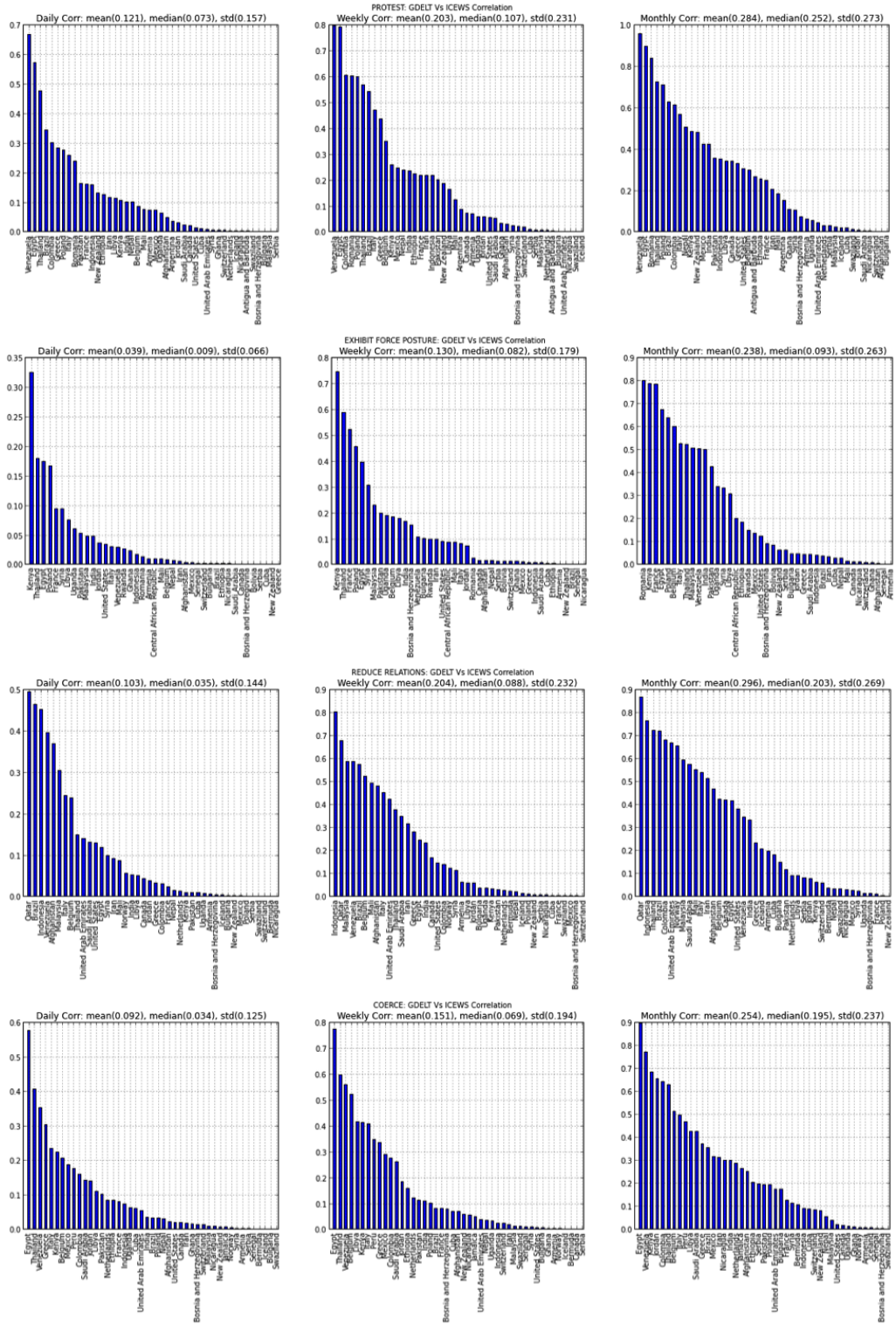

Figure 3.17: CAMEO Category 13 - 16

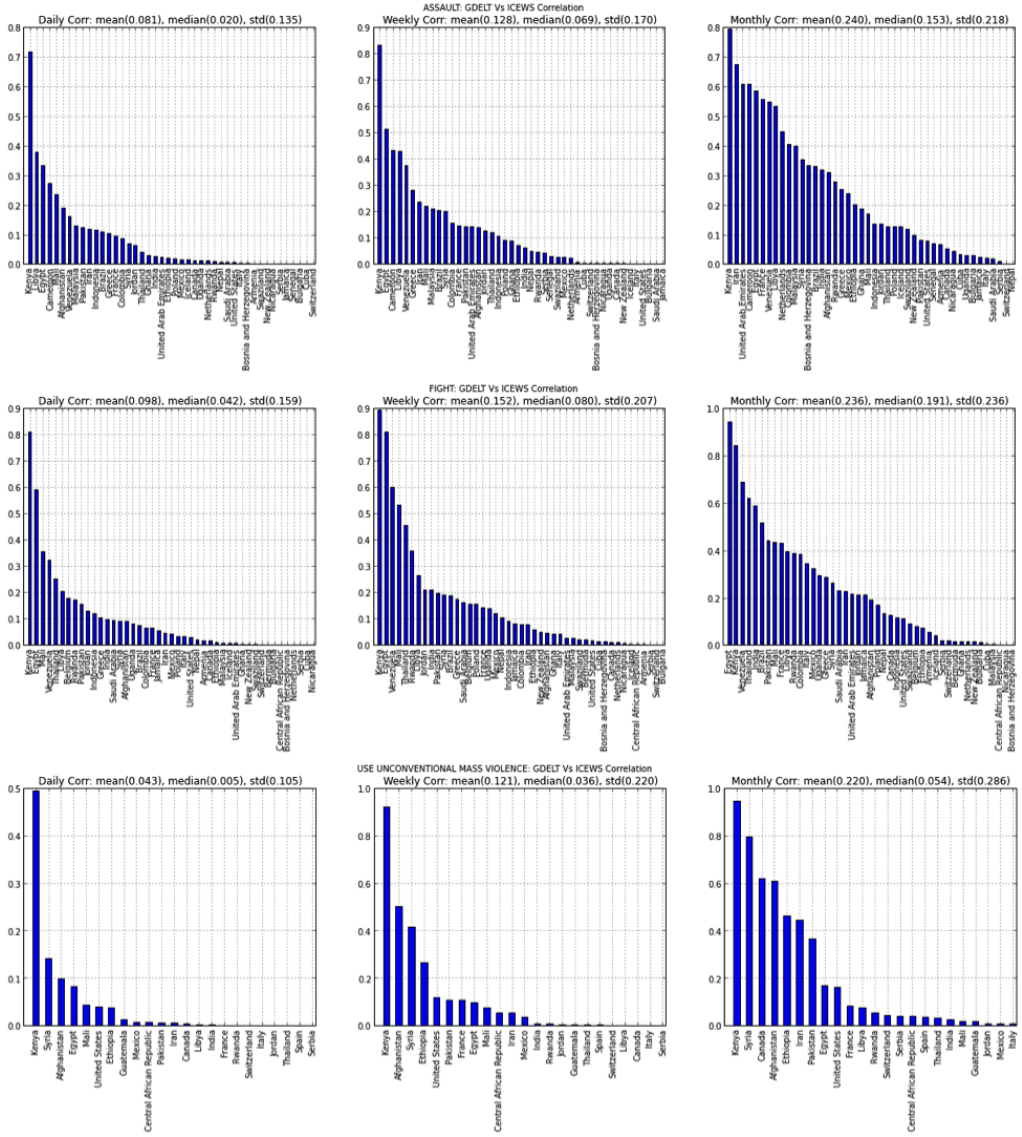

Figure 3.18: CAMEO Category 17 - 20

3.5.2 Correlation Analysis (Militarized Interstate Disputes [MIDS])

As a further check of the reliability of GDELT and ICEWS in the other event categories, we compare the reported events of these event data sets against the long-standing Militarized Interstate Dispute (MIDB) dataset [90]. This dataset is currently in its fourth full iteration, covering major interstate disputes in several categories from 1816 to 2010. This dataset has been a core dataset in international relations for more than two decades and has undergone thorough revision throughout.

We utilize MIDB 4.01 version of the data, which records events on the participant level, with one record per militarized dispute. We kept only the cases in which the reported participant originated the dispute. The data also provided a start and end date for each dispute.

There are three areas of overlap between the MID dataset and the CAMEO coding framework of GDELT and ICEWS: (1) threats - 13 in CAMEO ("threaten") and 2 in MIDs ("threat of use of force"); (2) display force - 15 in CAMEO ("exhibit force posture") and 3 in MIDs ("display of force"); and (3) use of force - 17, 18 and 19 in CAMEO ("coerce", "assault", and "fight") and 4 and 5 in MIDs ("use of force" and "war"). Since we are using the origin countries from MIDB, this is compared with Actor 1 (or the source actor) in GDELT and ICEWS.

There are three expectations. (i) Since MIDB records the most prominent events of these types, we would expect many more reports of such events during the months in which militarized interstate disputes (MIDs) are reported. (ii) For a similar reason, we would expect the proportion of months in which an incident is reported by GDELT and ICEWS to approach 1 during months in which a MID is ongoing. (iii) We expect much lower counts and proportions during periods in which MIDs are not reported. Since we are measuring these at the country-month level, we would expect this to be a relatively easy test for the computer-generated event datasets.

Table 3.12 reports the results comparing all the GDELT events to the MIDB dataset. The results, as we might expect, are quite mixed. Comparing any reported events in GDELT against the months in which any type of comparable MID is reported seems to suggest some success. The number of reported events in MID months is indeed higher than in months where MIDs were not recorded. With that being said, GDELT only picks up MID events in

about 40% of MID months. Breaking this down by category, GDELT does a much better job picking up acts of force than threats or displays of force – which is not surprising, since these are the ones likely to generate greater attention. Table 3.13 confirms this for root events as well. Put simply, as with the protest category, there is some signal to the computer-generated event data, but it also seems to be missing quite a bit of information about the duration of events, even when measured at the monthly level.

Table 3.12: Comparison of All GDELT Events and MIDS Datasets

All Events	**GDELT >0**
MID = 1	Avg. Count = 2.30
	Pct. Report = 0.40
MID = 0	Avg. Count = 0.61
	Pct. Report = 0.17
Threats	**GDELT >0**
MID = 1	Avg. Count = 0.45
	Pct. Report = 0.15
MID = 0	Avg. Count = 0.25
	Pct. Report = 0.10
Displays	**GDELT >0**
MID = 1	Avg. Count = 0.09
	Pct. Report = 0.05
MID = 0	Avg. Count = 0.03
	Pct. Report = 0.02
Use of Force	**GDELT >0**
MID = 1	Avg. Count = 1.51
	Pct. Report = 0.33
MID = 0	Avg. Count = 0.48
	Pct. Report = 0.15

Table 3.14 conducts a similar test for ICEWS. Again, ICEWS seems to have some signal for shows of force, but does much worse with threats and displays of force. Indeed, with threats, the number of articles recording threats is higher in non-MID months than in MID months. ICEWS also fails to record any MID-related event in several months in which the

Table 3.13: Comparison of GDELT Root Events and MIDS Datasets

Root Events	**GDELT >0**
MID = 1	Avg. Count = 1.48
	Pct. Report = 0.33
MID = 0	Avg. Count = 0.41
	Pct. Report = 0.17
Threats	**GDELT >0**
MID = 1	Avg. Count = 0.28
	Pct. Report = 0.11
MID = 0	Avg. Count = 0.16
	Pct. Report = 0.10
Displays	**GDELT >0**
MID = 1	Avg. Count = 0.06
	Pct. Report = 0.04
MID = 0	Avg. Count = 0.02
	Pct. Report = 0.02
Use of Force	**GDELT >0**
MID = 1	Avg. Count = 0.97
	Pct. Report = 0.27
MID = 0	Avg. Count = 0.33
	Pct. Report = 0.15

MIDs dataset records a MID event.

All of these results suggest, again, that there is some signal in the computer-generated event data, but they are far from being either completely reliable in picking up events when they are happening or in not recording events when they are not occurring. This was a relatively easy test, given that the analysis occurred at the month level and the MIDs dataset is likely to only cover the most prominent examples of such disputes.

Table 3.14: Comparison of ICEWS Events and MIDS Datasets

All Events	**ICEWS >0**
MID = 1	Avg. Count = 2.58
	Pct. Report = 0.34
MID = 0	Avg. Count = 0.22
	Pct. Report = 0.09
Threats	**ICEWS >0**
MID = 1	Avg. Count = 0.08
	Pct. Report = 0.06
MID = 0	Avg. Count = 0.11
	Pct. Report = 0.04
Displays	**ICEWS >0**
MID = 1	Avg. Count = 0.012
	Pct. Report = 0.011
MID = 0	Avg. Count = 0.004
	Pct. Report = 0.003
Use of Force	**ICEWS >0**
MID = 1	Avg. Count = 2.22
	Pct. Report = 0.33
MID = 0	Avg. Count = 0.21
	Pct. Report = 0.08

3.5.3 Duplication Analysis for All Cameo Categories

To analyze the duplication of GDELT and ICEWS events for other CAMEO categories, we collected all GDELT events with corresponding news articles during the period April 2013 to May 2014 and full set of ICEWS events in the same period. The duplication rates of each category for ICEWS and GDELT are both between 10% to 30%, which is consistent with our above duplication analysis over protest events. Figure 3.19 shows the duplication rate for each categories. This suggests that there is no improvement in de-duplication in the non-protest event codes.

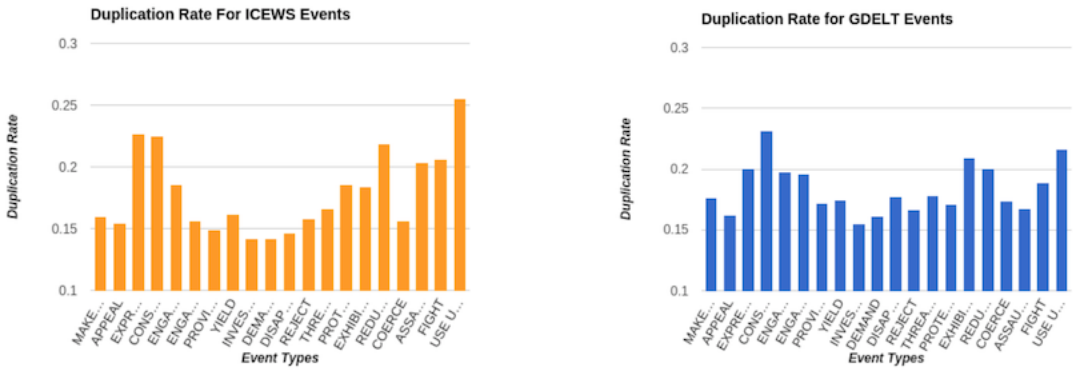

Figure 3.19: Duplication rate for ICEWS and GDELT Events

3.5.4 Analysis of Event Coding Quality

While we were limited by time constraints in our ability to qualitatively evaluate the validity of a large number of examples from other event categories, we did attempt to look at a large number of examples. To do this, we constructed a program on Amazon Web Services which displayed the GDELT coding of the events and the associated article used by GDELT. Three graduate students and two of the co-authors embarked on checking whether the type of event reported matched the type of event in the article and if it occurred within a reasonable window around the reported date. We did not check the location, source country or target country. Thus, we are providing a relaxed test of the validity of these events. Figure 3.20 shows what the program looks like. To improve human coding accuracy, we grouped the examples by event code, so that coders would see consecutive examples of the same CAMEO category and would not need to memorize the entire sequence.

Note: The Figure 3.20 also gives an example of errant coding. The events described in this article do suggest engagement in ethnic cleansing, but the events described are from the fictional Left Behind series - a set of books about the apocalypse based on The Book of Revelations in The Bible.

It became immediately clear that there were issues throughout the event categories - including severely misreported timing of events, lack of context, and other issues like those reported in the protest category. For example, in the "Use Unconventional Mass Violence" category there is a severe problem with the reported timing - most of the events reported as

Figure 3.20: Event Tagging Program

occurring in 2014 occurred much earlier. This makes sense, since mass violence is relatively rare, but news outlets regularly mention past instances of mass violence. We found references to events from World War II, Vietnam, the Stalinist period in the Soviet Union, the 1990s civil war in Yugoslavia, etc. Only a few cases were actual uses of mass violence within the 2-year period of when GDELT recorded it. Even more strangely, the category "Make Optimistic Statement" seems to pick up on job advertisements on a regular basis.

We randomly chose 150 example articles from all of the 20 CAMEO categories in GDELT, reported during the period between March and May 2014, for a total of 3,000 articles. Overall, 4,100 events were coded, so several of the articles were classified by more than one coder. When events were coded by more than one coder, they agreed 75% of the time, with the final label decided by majority rule. When an article was classified by two coders who disagreed, one of them was chosen at random. All of the 3,000 sampled articles were classified by at least one coder.

Figure S14 shows the summary statistics for the different classes. The average accuracy is about 16.2%. The most accurate category was protest events, with about 35.3% accurately

classified. Some of the categories, such as “Express Intent to Cooperate” (03), “Reject” (08), and “Use Unconventional Mass Violence” (20) have less than 10% accuracy. We hesitate to say that these are firm numbers, given the low number of total cases sampled, especially in certain categories, but it is certainly discouraging for anyone who might suspect that the problems identified above are limited to protest events.

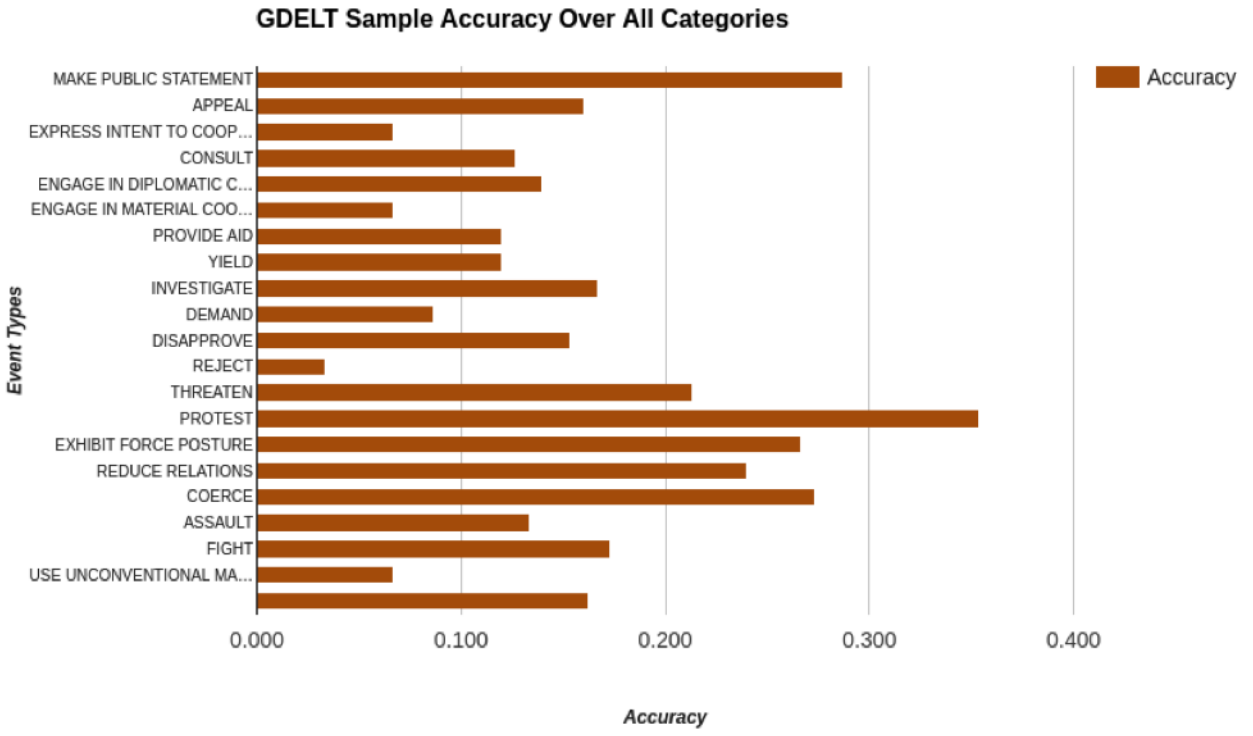

Figure 3.21: Accuracy of GDELT Coding Across Multiple Categories

3.6 Discussion

There have been serious efforts over the past 40 years to use newspaper articles to create global-scale databases of events occurring in every corner of the world, in order to help understand and shape responses to global problems. Although most have been severely limited by the technology of the time , two recent groundbreaking projects to provide global, realtime âĂIJevent dataâĂİ that take advantage of automated coding from news media have gained widespread recognition: International Crisis Early Warning System (ICEWS),

maintained by Lockheed Martin, and Global Data on Events Language and Tone (GDELT), developed and maintained by Kalev Leetaru at Georgetown University [67]. The scale of these programs is unprecedented, and their extraordinary promise has been reflected in the attention they have received from scholars, media, and governments. However, as we demonstrate below using newly available data, they suffer from major issues with respect to reliability and validity. Opportunities exist to use new methods and to develop an infrastructure that will yield robust and reliable "big data" to study global events from conflict to ecological change [67].

Automated event coding parses individual sentences into SUBJECT VERB OBJECT format and categorizes the action using a framework like CAMEO (Conflict and Mediation Event Observations). So a statement like "Secretary of State John Kerry complained about Russia's support of Syria's Ba-shar al-Assad" would be coded as US GOVERNMENT/DISAPPROVE/RUSSIAN GOVERNMENT. This can then be further refined into a numeric level of hostility or cooperation by using scales like the Goldstein Score. Whereas CAMEO focuses on categories for international and domestic conflict, similar frameworks could be developed for almost any kind of interaction in news media (e.g., transactions between businesses or debates over scientific findings).

Uses for the resulting data have been manifold. In international relations, hand-coded and automated event data have been used to anticipate conflict escalation [107]. When combined with statistical and agent-based models, ICEWS claims a forecasting accuracy of 80%. GDELT has been used to track, e.g., wildlife crime and the rise of hate speech following the U.K. Brexit vote.

There are several challenges in the current approach. First, the focus on sentences removes a great deal of context. Event occurrences do not neatly partition into sentences. This lack of context, for example, often fails to distinguish rereporting of historic events, and this results in high rates of duplication.

Second, event data programs can have inconsistent corpuses over time. For instance, GDELT has expanded the number and variety of its sources. Although expansions are generally positive-incorporating, for example, more non-Western news sources-they result in difficulty interpret-ing what a spike in GDELT data at a particular time means; the project has not been entirely transparent on how these expansions have taken place. ICEWS has been more

consistent about maintaining a common set of sources across nearly 25 years.

Third, the text-processing systems used in event coding are still similar to ones de-veloped more than 20 years ago. Although ICEWS has recently begun leveraging a machine learning approach, GDELT still relies on dictionary-based pattern matching that leads to overly simplified or misclassified coding instances. The field of text processing has developed a range of tools to address these issues [49]. Finally, although there are a few large event-coding programs, the academic groups working on these problems are surprisingly diffuse and isolated.

RELIABILITY Our first set of experiments deals with the reliability of event data, whether programs ostensibly using similar coding rules produce similar data. We used four sources of event data [ICEWS, GDELT, Gold Standard Reporgrot (GSR), and Social, Po-litical and Eco-nomic Event Database (SPEED)], all designed with the capability to detect protest events. GDELT and ICEWS are fully automated and are the best attempts so far at realtime global event data. The GSR data set, generated by the nonprofit MITRE Cor-poration, is hand-coded from local and international newswires in Latin America since 2011 [94]. SPEED is a semiautomated global event data system by the University of Illinois that uses a combination of human and automated techniques for identifying events. It touts the high validity of its event coding [44]. GSR and SPEED were developed to provide a "ground truth", but their methods would be difficult and expensive to scale. Although these systems have different origins (e.g., ICEWS was meant to encode strategic interactions, often among nation-states, and GSR was meant to focus on tactical, local issues), we anticipate that overall there should be a high correlation between the time series of events generated by these projects, even if the event counts are not comparable.

We find that the correlation between event data collections is in the area most consider âĂIJweakâĂİ [correlation coefficient (r) < 0.3]. The average correlation between GDELT and the GSR across Latin American countries is 0.222, and the correlation between ICEWS and the GSR is 0.229. SPEED and GDELT records only match (i.e., both data sets recorded a protest happening on the same day) 17.2% of the time. SPEED and ICEWS only agree on 10.3% of events. ICEWS and GDELT also rarely agree with each other, producing an average correlation across Latin American countries of 0.317.

These correlations vary dramatically between countries and improve when there are large upticks in event counts. For example, the large uptick in protests in Venezuela in January 2014 is well captured by both ICEWS and GDELT. They also improve when the time scales are made rougher (from daily to weekly or monthly). Reliance on English language news coverage results in stronger correlations for states that receive more coverage in the Western press (e.g., Brazil) [118].

VALIDITY To assess the validity of event-coding projects-the degree to which they reflect unique real world events, we leveraged a special characteristic of the GDELT data set. Since its launch on 1 April 2013, GDELT has provided URLs for most of its coded events. We looked at all protest events up to 2 July 2014 (431,549 records), extracted content for records with a valid URL (344,481 records), and filtered them to assess the validity of their classification as protest events. This yielded 113,932 unique, nonduplicated events, articles that are highly probable to be about protests at the time reported.

Even for these filtered records, only 49.5% are classified as referring to actual protests, roughly in line with what we found in 1000 human-coded records. After keyword and temporal filtering, de-duplication of events, and machine learning classification of real events from nonevents or planned events, only 21% of GDELT's valid URLs indicate a true protest event.

The ICEWS system was more robust (about 80% of keyword-filtered events were classified as protest events) but still vulnerable to duplicate events ($<$20% of the recorded events). The bottom line is that computer automated event data often duplicate and misclassify events, and there are tools, including ones used here, to deal with many of these issues. Similar tests for the other 19 event categories in GDELT and ICEWS found similar problems.

POLICY IMPLICATIONS Coding interactions in news media is com-plex, as it involves actor recognition and normalization, time-frame detection, geocoding, event encoding, and classification, multilingual support, and other issues. Yet the history of event data has generally been one of small teams and underfunded re-search. It has not helped that much of the development has taken place in political sci-ence, a discipline under constant threat of having its National Science Foundation (NSF) funding cut by Congress.

As scholars and government agencies create the next generation of large-scale event data, two goals should shape their efforts. First, new efforts must develop a multidisciplinary community of scholars, including computational linguists, data analytics professionals, information extraction practitioners, and domain experts. Although there have been improvements in the natural language processing used for ICEWS (9), innovation in the event data field has generally been slow, especially in handling contextual features and temporal and geographic information. Neither ICEWS nor GDELT were designed to try to deduplicate events; in fact, multiple occurrences have sometimes been used to denote event significance and to support improved modeling. The one sentence per event model is not sufficient for predictive, diagnostic, or abductive reasoning. Research on probabilistic databases can help one reason about inconsistency issues in information extraction and how best to integrate imprecise information into event coding (10, 11). It is time to develop a strong community of teams competing to create the best possible event data, and their event-coding software should, ideally, be released publicly to encourage community engagement.

Second, the corpus used to create event data must be made explicit, and, to the greatest extent possible, shared between teams. As demonstrated by legal issues faced by GDELT [a dispute over use of source materials resulted in major scholars abandoning the project and obstacles to us-ing the data for publication], the current system, where corpus development can only be done by well funded individual teams with exclusive rights to material, is deeply problematic and encourages atomization of the field. Such a corpus should include more non-English sources to avoid some of the issues observed above.

We recommend development of open test beds of event data against which different approaches can be tested. These test beds should be composed of a representative set of textual data, where some portion has been carefully hand-coded. Such a test bed can be used in contests, along the lines of those sponsored by DARPA (Defense Ad-vanced Research Projects Agency) or TREC (Text Retrieval Conference), where different approaches to text analysis compete to produce the best automated coding for event data. This would allow scholars to test tools already developed for text analysis in other areas of study and to produce new tools to deal with the specific problem of tracking interactions from media reports.

A consortium should be developed to provide real-time controlled access to a comprehensive

array of copyrighted material, to protect the business interests of news agencies, and to elicit broader social interest in event data. The UN Global Pulse initiative and Flowminder in Sweden, which address similar issues with regard to cell phone data, could provide a model.

Programs like that proposed here have been tried in other areas, such as social media analysis and search-engine technology, with strong results. Such an effort can go a long way toward settling the debate over the extent to which fully automated approaches, like those of GDELT and ICEWS, can compete with semiautomated approaches like that of SPEED.

Event data can provide insights into a range of global problems such as national security, economic instability, environmental concerns, and the spread of diseases. Our ability to reason about world affairs would be significantly improved by the availability of highquality event data.

Chapter 4

Event Detection and Key Information Extraction

4.1 Introduction

Identifying and extracting relevant information from large volumes of text articles play a critical role in various applications ranging from question answering [119], knowledge base construction [115] and named entity recognition [101]. With the rise and pervasiveness of digital media such as news, blogs, forums and social media, automatically detecting the occurrence of events of societal importance, further categorizing them by performing event classification (i.e., type of event) and automatically/manually encoding them are popular areas of research. As a case in point, Muthiah et al. [83] use a dictionary-based approach to first identify news articles pertaining to planned civil unrest events, extract key indicators of these events and later use this information to predict the onset of protests. Other applications include event forecasting [95], social media monitoring [51] and detecting financial events [6].

In this study, we view the twin problems of event detection and extracting key sentences to enable event encoding and classification in a unified manner as a form of multiple instance learning (MIL) [29]. This enables us to identify salient event-related sentences from news articles without training labels at the sentence level and the use of manually defined dictionaries. Our motivation stems from the practical contexts in which event extraction systems

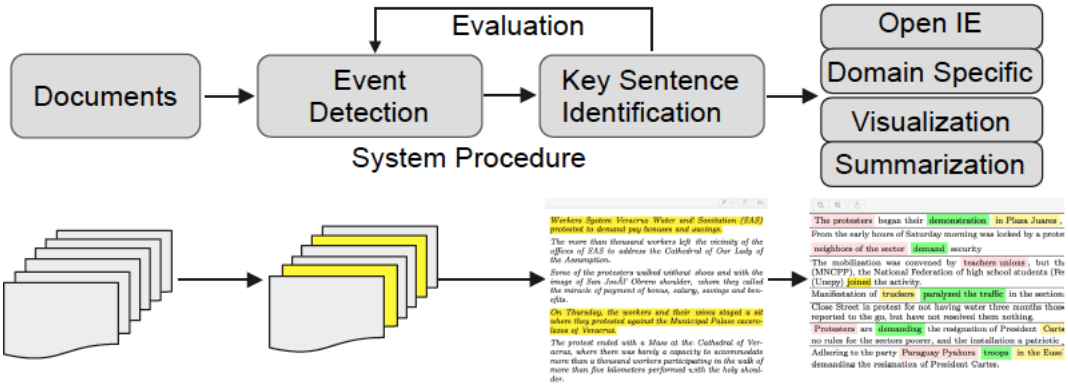

Figure 4.1: System Overview

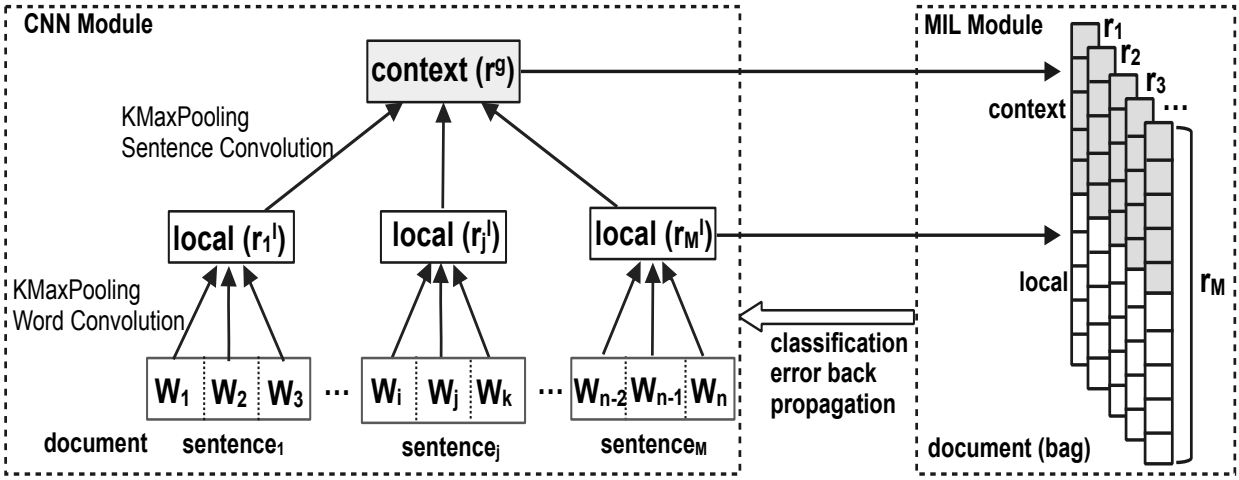

Figure 4.2: MI-CNN Model Overview.

operate (see Figure 4.1). In a typical news article, there exist a small set of key sentences that provide detailed information for a specific event. Identifying these sentences automatically is useful for succinctly summarizing the news article. Highlighting such key sentences within a visualization tool will enable a human analyst to quickly locate important information, rapidly scan the contents of the article, and make timely decisions. Additionally, as we will demonstrate, key sentences can form the basis of automated event encoding and we can extract the final event based on the identified salient sentences. Figure 4.1 provides an overview of the methods developed in this work.

In more detail, we propose an MIL approach based on convolutional neural networks (CNN) that incorporates a distributed representation of documents to extract event-related sen-

tences. Specifically, we consider each individual sentence within a document to be an instance and the collection of instances within a document as a bag. We are provided with labels at only the bag (document) level. A positive label indicates that the news article refers to a protest related event. Our model seeks to predict the labels at the document and sentence levels but with no available sentence-level labels during training. Traditional MIL formulations [29, 9, 73] treat each instance (sentence) within a bag (document) as independent of each other. Our model relaxes this strong assumption by combining local and global context information to construct a continuous sentence level representation. We evaluate our proposed model on the specific domain of civil unrest events such as protests, strikes and "occupy events", with data obtained from ten Latin American Countries.

The major contributions of our work can be summarized as follows:

- We propose a novel framework which views event detection and identification of key sentences as a form of multiple instance learning.
- We develop a novel sentence representation that combines local and global information using convolutional neural network formalisms.
- We propose a new MIL-based loss function that encourages selection of a small set of salient sentences for the protest articles.

4.2 Problem Definition

Given a set of N news articles, $\{x_i\}, i = 1..N$, each news article is associated with a label $y_i \in \{0, 1\}$ indicating whether the article refers to a protest event or not.

Our goals here are twofold. First, we aim to predict the event label $\hat{y}_i$ for each news article x_i. This is the standard text classification formulation for solving the event detection (recognition) problem. Our second goal is to extract a small set of salient sentences that are considered as indicative (key) of event related information. The dynamic number $k = |x_i| \times \eta$ of sentences to extract is decided by the length of the article, where η in $(0, 1]$ is a predefined value. We define the second task as **key sentences extraction** problem. The extracted key sentences are helpful for related tasks such as event detection, classification, encoding,

summarization, and information visualization.

4.3 Proposed Model

We propose a multiple instance learning (MIL) model based on convolutional neural networks (MI-CNN) for our task. Each text article is considered as a *bag* and sentences within the bag are individual instances. We have labels only for the article-level (bags) and do not have individual ground truth labels available for each sentence (instances). Similar to MIL formulations [73, 62], we seek to predict the document-level labels and transfer the labels from the bag-level to individual sentences to identify the key sentences summarizing the protest-related information.

We utilize CNN to construct a distributed representation for each instance (sentence), that are the input to the MIL framework. Using the feedback from MIL training process, the CNN module updates the instance representation. For every sentence within an article, our model estimates a sentence-level probability that indicates the belief of the sentence indicating event related information. The MI-CNN applies an aggregation function over the sentences to compute a probability estimate for an article referring to a protest. Figure 4.2 provides an overview of our proposed model.

4.3.1 Instance Representation

As seen in Figure 4.2, the raw word tokens from the article are input into the network. Given that a sentence s consists of D words $s = \{w_1, w_2, ..., w_D\}$, every word w is converted to a real value vector representation using a pretrained word embedding matrix W. The individual word representations are then concatenated for every sentence. The embedding matrix $W \in R^{d \times |V|}$, where d is the embedding dimension and V is a fixed-sized vocabulary, will be fine-tuned during the training process.

The first convolution and k-max pooling layer are used to construct the local vector representations for every sentence referred by $\mathbf{r_j}^l$ for the j-th sentence. The convolutional layer scans over the text, produces a local feature around each word and captures the patterns regardless of their locations. The k-max pooling layer only retains the k-most significant

feature signals and discards the others. It creates a fixed-sized local vector for each sentence.

Given a sentence, s, the convolution layer applies a sliding window function to the sentence matrix. The sliding window is called a kernel, filter, or feature detector. Sliding the filter over the whole matrix, we get the full convolution and form a feature map. Each convolution layer applies different filters, typically dozens or hundreds, and combines their results. The k-max pooling layer applied after the convolution layer output k values for each feature map. In addition to providing a fixed-size output matrix, the pooling layer reduces the representation dimensionality but tends to keep the most salient information. We can think of each filter as detecting a specific feature such as detecting if the sentence contains a protest keyword. If this protest-related phrase occurs somewhere in the sentence, the result of applying the filter to that region will produce a large value, and small values in other regions. By applying the max operator we are able to keep information about whether or not the feature appears in the sentence.

The local features, $\mathbf{r_j}^l$, aim to capture the semantic information embedded within the scope of the j-th sentence. These local representations are then transformed using another convolution and k-max pooling layer above to construct the article-level context representation, denoted by $\mathbf{r}^g$. The context features $\mathbf{r}^g$ capture the information across all the sentences within the article and are shared by all the sentences. For every sentence, it's specific local representation is concatenated with the context representation and used for the MIL-based optimization. This combined representation is denoted by $\mathbf{r_j}$ for the j-th sentence.

$$\mathbf{r_j} = \mathbf{r_j}^l \oplus \mathbf{r}^g, \tag{4.1}$$

where $\oplus$ is the concatenation operator. Intutively, the context feature vector encodes topic information of the document and is useful for distinguishing the theme and disambiguating polysemy encoded in local features, $\mathbf{r_j}^l$. For instance, a sentence containing the token *strike* may refer to a civil unrest event, but it is also often related to a military activity. Without context information, it is very hard to make this decision.

4.3.2 Sentence- and Document-Level Estimates

Given the distributed representation $\mathbf{r_j}^i$ of the j-th sentence in the document x_i, we compute a probabilistic score p_j^i using a **sigmoid** function:

$$p_j^i = \sigma(\theta^T \mathbf{r_j}^i + b_s) \tag{4.2}$$

where θ is the coefficient vector for sentence features and b_s is the bias parameter. Intuitively, p_j^i is the probability that the j-th sentence within article x_i refers to information pertaining to a protest. Aggregating these estimated probabilities over these indicative sentences will provide an estimate for a document to indicate a protest event. To alleviate the bias of varying lengths of different articles, we choose a predefined ratio, η (set to 0.2), to choose the dynamic number of key sentences. We choose the set of top highly ranked sentences K_i as key sentences in each article x_i. $|K_i| = max(1, \lfloor |x_i| \times \eta \rfloor)$. Generally, we will select one or two sentences each article given η as 0.2 in our dataset.

We compute the probability P_i of an article referring to a civil unrest event as the average score of the key sentences:

$$\text{Prob}(y_i = 1) = P_i = \frac{1}{|K_i|} \sum_{k \in K_i} p_k^i \tag{4.3}$$

There are several other common options to aggregate the instance probabilities to bag-level probability. Averaging is one of the most common aggregation functions. It is suitable for the cases where the bag label is decided by majority rule. Another common option is the max function. In this case, the bag probability is decided by the most significant instance. Noise-OR is also a aggregation function used often in MIL. It tends to predict bags to be positive due to its natural property. In protest news articles, there often exists a small set of sentences indicating the occurrence of a protest event and remaining sentences are often related to the background or discussion about that event. In this case, using the average over all sentences makes the salient sentences indistinguishable from the background sentences. However, using the *max* function makes the model sensitive to longer documents. We ran preliminary experiments based on these different aggregation functions.

4.3.3 Multiple Instance Learning (MIL)

During training, the input to the MIL module is a document x_i consisting of individual sentences; label $y_i \in \{0, 1\}$ provided for the document. To encourage the model to select meaningful key sentences, we design a compositional cost function that consists of four components: bag-level loss, instance ratio, instance-level loss, and an instance-level manifold propagation term. The loss function is given by:

$$
\begin{aligned}
L(x, y; \theta, W, F, b) = & \underbrace{\frac{1}{N} \sum_{n}^{N} (1 - y_n) \log P_n + y_n \log (1 - P_n)}_{\text{bag-level loss}} \\
& + \underbrace{\frac{\alpha}{N} \sum_{n}^{N} y_n \max(0, |K_n| - Q_n) + (1 - y_n) Q_n}_{\text{instance ratio control loss}} \\
& + \underbrace{\frac{\beta}{N} \sum_{n}^{N} \frac{1}{M_n} \sum_{m}^{M_n} \max(0, m_0 - \text{sgn}(p_m^n - p_0) \theta^T r_m^n)}_{\text{The instance-level loss}} \\
& + \underbrace{\frac{\gamma}{(\sum_n M_n)^2} \sum_{n}^{N} \sum_{i}^{N} \sum_{m}^{M_n} \sum_{j}^{M_i} (p_m^n - p_j^i)^2 e^{(-\|r_m^n - r_j^i\|_2^2)}}_{\text{instance-level manifold propagation}}
\end{aligned}
$$

where $Q_n = \sum_m 1(p_m^n > 0.5)$ is an indicator function that returns the number of instances with a probability score greater than 0.5. N is the number of documents and M_n is the number of sentences in n-th document. Hyper-parameters α, β, and γ control the weights of different loss components. Dropout layers are applied on both word embedding and sentence representation to regularize the model, and Adadelta [126] is used as the model optimization algorithm. We used a dropout rate of 0.2 in the word convolutional layer and 0.5 in the sentence convolutional layer. α, β and γ are 0.5, 0.5 and 0.001, and are chosen by cross-validation on training set, respectively.

- **Bag Level Loss**: this component is the classical cross-entropy loss for classification

which penalizes the difference between predictions and the true labels for bags.

- **Instance Ratio Control Loss**: this component encourages no sentence in the negative article to have a high probabilistic estimate and pushes the model to assign high probabilistic estimates to a smaller set of sentences in the positive articles.
- **The Instance-Level Loss**: this part is a standard hinge loss that encourages wider margin (m_0) between positive and negative samples. Here sgn is the sign function. The hyper parameter m_0 and p_0 control the sensitivity of the model. p_0 determines positiveness of instance. We set m_0 as 0.5 and p_0 as 0.6 in our case.
- **Instance-level Manifold Propagation**: Inspired by [62], the manifold propagation term encourages the similar sentence representations to have similar predictions/estimates.

To optimize the cost function we use mini-batch stochastic gradient descent. This approach was found to be scalable and insensitive to the different parameters within the proposed model. A backpropagation algorithm is used to compute the gradient in our model. In our experiments, the MI-CNN model was implemented using the Theano framework [12].

4.4 Experiments

4.4.1 Dataset

In our experiments, we use a manually labeled dataset (GSR; Gold Standard Report) of Spanish protest events from ten Latin America countries [1] from October 2015 to Jan 2016. The dataset consists of 19795 news articles that do not refer to a protest (negatives) and 3759 articles that are protest-related (positives). For each positive article, the GSR provides the population and event type of the protest event. The event population indicates the type of participants involved in the protest. The event type identifies the main reason behind the protest. The set of event population and event types are listed in Table 4.1. Each annotated sample is checked by three human analysts and the labels are confirmed if two of them agree on the assignment. We use 5-fold cross validation for evaluation. On average, we have 18844

[1] Argentina, Brazil, Chile, Colombia, Ecuador, El Salvador, Mexico, Paraguay, Uruguay and Venezuela

Table 4.1: Event population and Type

Event Population
General Population
Business
Legal
Labor
Agricultural
Education
Medical
Media

Event Type
Government Policies
Employment and Wages
Energy and Resources
Economic Policies
Housing

articles for training and 4710 for test in each fold. Since the dataset is imbalanced we report precision, recall, and F1 score computed for the positive class for our experiments.

During the data pre-processing phase, we augment a special token (*padding*) by $\frac{T-1}{2}$ times to the beginning and the end of the sentence, where T is the window size of filter in word convolution layer. For the mini-batch setting in Theano, we define two variables max_s and max_d to control the maximum number of tokens for each sentence and maximum number of sentences for each document. The special token (*padding*) is appended to the end of each sentence until max_s is achieved. Likewise, the *padding* sentences are attached to the end of a document until max_d achieved. We set max_s as 70 and 30 for max_d in our experiments.

Pretrained Word Embedding For the initial word embeddings, we use a training corpus consisting of 5.7 million Spanish articles ingested from thousands of news and blog feeds covering Latin America area during the time period of Jan 2013 to April 2015. The open source tool *word2vec* [80] is used for pretraining word embeddings in our experiments. We set the word embedding dimension as 100. Tokens appearing less than ten times are removed and we use the skip-gram structure to train the model.

4.4.2 Comparative Methods

Support vector machines (SVM) are known to be effective for the standard text classification problem [53, 112]. We use SVM as one of the baseline models for the article classification problem. We remove Spanish stop words, apply lemmatization on tokens, and use TF-IDF features.

The second comparative approach used in our study is a CNN with a softmax classifier. The CNN model first constructs a sentence vector by applying convolution and k-max pooling over word representations. Then a document vector is formed over sentence vectors in a similar way. Finally, the softmax layer uses the document vector as input and predicts the final label.

Although the SVM and CNN model can classify whether an article refers to a protest or not, they do not directly output the key sentences referring to the events. Both SVM and CNN models construct a document level representation (global information) and use it as input to final classifier; we refer to them as global methods.

Table 4.2: Hyperparameters for MI-CNN model

N	Batch size	50
max_w	Max number of words in sentence	70
max_s	Max number of sentences in a article	30
f_w	Number of feature maps in word Conv layer	50
f_s	Number of feature maps in sentence Conv layer	100
k_w	k-max pooling parameter in word Conv layer	3
k_s	k-max pooling parameter in sentence Conv layer	2
T_w	Filter window size in word Conv layer	5
T_s	Filter window size in sentence Conv layer	3
η	The ratio of choose key sentences	0.2
α	The control parameter of Instance ratio control loss	0.5
β	The control parameter of Instance level loss	0.5
γ	The control parameter of Instance level manifold propogation	0.001
$drop_w$	Dropout rate in word Conv Layer	0.2
$drop_s$	Dropout rate in sentence Conv Layer	0.5
d	Pretrained word embedding dimension	100

As opposed to global methods, local methods assign credit to individual sentences and make the final decisions based on an aggregation function applied over the individual sentences. As such, these approaches can extract the set of significant sentences along with an article-level label prediction. The multiple instance support vector machine (MISVM) [32], group instance cost function (GICF) [62] (discussed in Section 4.4.3), and our proposed approach (MI-CNN) are all local methods. To train the GICF and MISVM models, we use the sentence representation learned from the CNN model as instance features. Table 4.2 shows the hyper-parameters used in the model MI-CNN.

Table 4.3: Event detection performance. comparison based Precision, Recall and F-1 score w.r.t to state-of-the-art methods. The proposed MI-CNN method outperform state-of-the-art methods

	Precision(Std.)	**Recall(Std.)**	**F1(Std.)**
SVM	0.818 (0.019)	0.720 (0.008)	0.765 (0.009)
MISVM	0.724 (0.030)	0.584 (0.017)	0.646 (0.018)
CNN Model	0.732 (0.033)	0.783 (0.026)	0.756 (0.007)
GICF	**0.833 (0.019)**	0.421 (0.09)	0.553 (0.086)
MI-CNN (max)	0.685 (0.030)	0.730 (0.029)	0.706 (0.018)
MI-CNN (avg)	0.731 (0.069)	0.789 (0.042)	0.759 (0.026)
MI-CNN (context + k-max)	0.742 (0.036)	**0.813 (0.041)**	**0.775(0.006)**

4.4.3 Experimental Results

Event Detection (Article Classification)

Table 4.3 shows the classification results for MI-CNN and comparative approaches for identifying whether a news article is "protest-related" or not. We report the mean precision, recall and F1 score along with standard deviation across five folds. The MI-CNN approach outperforms all other baseline methods. Both MISVM and GICF models have relatively poor performance on this dataset. Specifically, the MI-CNN model outperforms GICF by 40% and MISVM by 20% with respect to the F1 score. One possible explanation for the poor performance of MISVM and GICF is that the sentence vectors learned from CNN model only capture the local information (sentence level) but ignore the contextual information important for article classification. In contrast to the GICF model, which uses fixed sentence representation learned from the CNN model, MI-CNN updates the sentence representation during the training process according to the feedback from the multiple instance classification.

Importance of Context Information To show that context information is helpful when encoding the sentence representation, we performed a set of experiments based on the vari-

ants of our MI-CNN model. We trained a model referred by MI-CNN (max), which does not add context information to the sentence vector. The maximum score of sentences in the article is used as the probability of an article to be positive. Different from MI-CNN (max) model, the second variant MI-CNN (avg) model infers the probability of a positive article as the average score over all the sentences. In the model referred by MI-CNN (context + k-max), the context information encoded into the sentence level representation and the dynamic top "k" sentences are used to infer the probability a given article to be positive (i.e., protest).

As shown in Table 4.3, the MI-CNN (max) model has worse performance when compared with SVM, CNN and two other MI-CNN models, which all use the global information to some extent. This experiment shows that exclusively using the local information is not beneficial for the classification task. Further, MI-CNN (context + k-max) achieves the best performance confirming the importance of context information.

Probability Distributions Figure 4.3 presents the distribution of the estimated document level probability estimates for protest and non-protest articles based on the aggregation of key sentence-level probability estimates. Within the MIL formulation, the sentence-level (instances within each bag) loss function attempts to separate the margin between the positive and negative sentences. The results show the stability of our predictions, because the majority of estimated probabilities for the protest articles are greater than 0.8, whereas for the non-protest articles are smaller than 0.2.

Identifying Key Sentences

In addition to classifying whether an article is reporting a civil unrest event or not, our model also extracts the most indicative sentences for each article. We perform a qualitative and quantitative evaluation of the indicative sentences.

Quantitative Evaluation

Since we do not have available ground truth data for the key sentences, we evaluate the quality of our identified sentences by comparing with sentences selected by several methods. We assume that key sentences should be discriminative about protest references. If we only use the selected sentences to represent the whole document and apply an article label

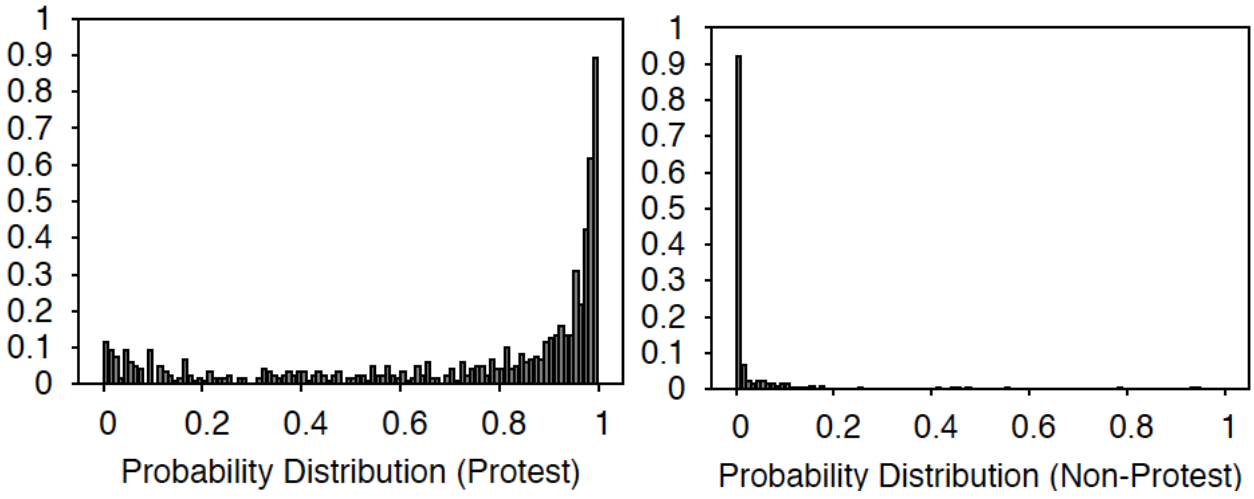

Figure 4.3: The histogram of probability estimates for protest and non-protest articles for test set

classifier on these documents, we expect that the selected sentences with higher quality will have better classification performance. In our experiment, we try three other methods for extracting the same number of sentences and apply the SVM classifier. The first baseline method (**Random**) randomly chooses sentences from a given article. News articles generally organize important information at the start and end of a document. As such, we select the first $\frac{k}{2}$ sentences from the start and $\frac{k}{2}$ from the end of an article as another baseline (**Start/End**). The third method (**Keywords**) selects sentences containing protest-related keywords such as *demonstration, march, protest* based on an expert-defined dictionary.

Table 4.4 shows the comparative results of the above outlined approaches. The MI-CNN approach outperforms all other methods with respect to F1 score. As expected, all methods show better performance than randomly choosing sentences. Using the sentences with protest-related keywords has the highest recall. However, this approach has a higher chance of false positives due to polysemy. For example, the term *march* can refer to the protest movement as well as the month of year. A significant strength of our proposed model compared to the keyword approach is that our model does not require any domain experts to curate a dictionary of protest keywords and is easier to adapt to new and unknown domains with minimal effort.

In addition to using the classifier to evaluate the quality of the sentences extracted by our model, we randomly choose 100 protest articles represented by key sentences for manual

Table 4.4: Event detection performance using key sentences only.

	Prec.(Std.)	Recall(Std.)	F1(Std.)
Keywords Protest	0.755 (0.021)	**0.638 (0.017)**	0.692 (0.018)
Random Sentences	0.681 (0.026)	0.433 (0.019)	0.551 (0.018)
Start/End Sentences	0.751 (0.022)	0.555 (0.026)	0.638 (0.019)
MI-CNN	**0.761 (0.015)**	0.635 (0.024)	**0.693 (0.019)**

evaluation. We ask three annotators to determine whether the extracted sentences refer to a protest event. If the sentence contains the participant and protest action information, we consider that the method correctly identified a sentence referring to a protest event. In case of inconsistencies amongst the human evaluators, the final decision is decided by a simple majority. The annotators agreed with each other 95% of the time in our labeling process. Figure 4.4 presents this human-based evaluation result. Our model has the highest average accuracy and least variance. The average accuracy that our model achieves is approximately 10% higher than **keywords** approach and 80% than **Start/End** approach.

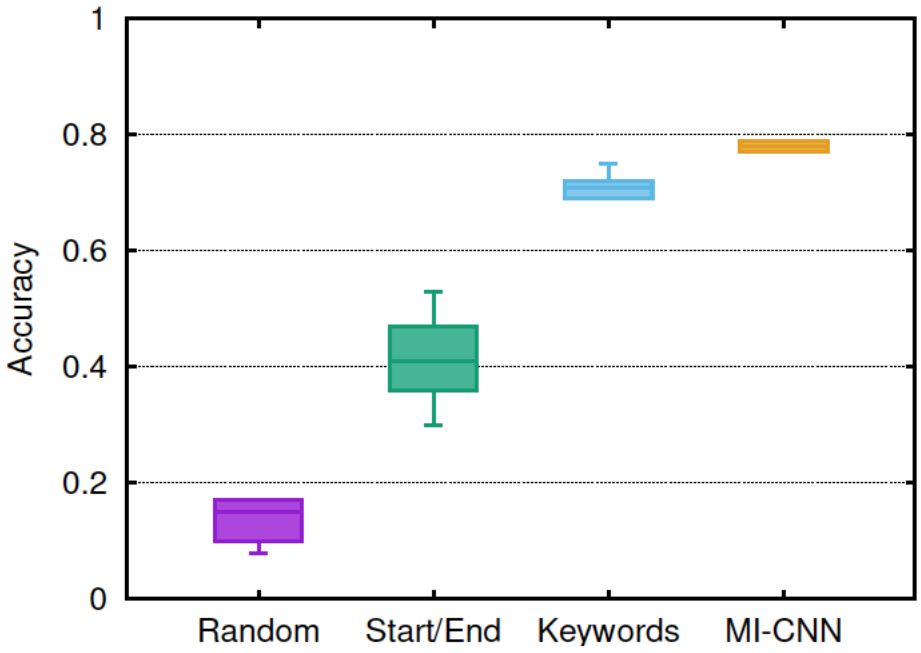

Figure 4.4: Event Reference Accuracy for Protest Articles

Qualitative Evaluation and Case Studies

Table 4.5: List of positive and negative sentences selected by our model sorted by score: The positive sentences show common patterns that include location references and purpose-indicating terms. The negative sentences may contain protest keywords, but are not related to a specific civil unrest event. The third and forth columns show whether the titled methods also select the same sentence as our approach as the key sentence. The pink color highlights the protest participant, green for protest keyword and yellow for location

Positive Sentences	Score	Keywords	Start/End
The protesters began their demonstration in Plaza Juarez, advanced by 16 September to Hidalgo.	0.9992	Yes	No
From the early hours of Saturday morning was locked by a protest the Francisco Fajardo highway from Caricuao, neighbors of the sector demand security	0.9991	Yes	No
The mobilization was convened by teachers unions, but the national March of public colleges and private (MNCPP), the National Federation of high school students (Fenaes) and the Center Union of secondary students (Unepy) joined the activity.	0.9991	No	No
Manifestation of truckers paralyzed the traffic in the section clean- Roque Alonso	0.9991	Yes	Yes
Close Street in protest for not having water three months those who protested pointed out that the problem was reported to the go, but have not resolved them nothing.	0.9991	Yes	Yes
Protesters are demanding the resignation of President Cartes, since they consider that - as they understand - no rules for the sectors poorer, and the installation a patriotic junta in power.	0.9991	Yes	No
Adhering to the party Paraguay Pyahura troops in the Eusebio Ayala Avenue heading to downtown Asuncion, demanding the resignation of President Cartes.	0.9991	No	Yes
From 09:00 hours, tens of inhabitants of the municipal head were concentrated at the entrance of Arcelia and almost 10 o'clock began a March toward the Center, which showed banners against staff of the PF.	0.999	Yes	No
A group of taxi drivers protested this Monday morning in the central town of el Carrizal municipality, in Miranda State, according to @PorCarrizal the demonstration is due to that, he was denied the circulation to the drivers who benefited from the transport mission.	0.9988	Yes	Yes
Negative Sentences	**Score**	**Keywords**	**Start/End**
Bled some guardians, also protesters, friends and family that went with them.	0.172	Yes	No
The parade by the 195 years of independence of Ambato yesterday (November 12) had a different connotation.	0.0125	Yes	No
This morning, the situation is similar, as already record barricades and demonstrations in the same place, by what police is already around the terminal.	0.0109	Yes	No
The young man asked that they nicely other costume to so participate in the parade.	0.0097	No	No
Employees announced that they will be inside until you cancel them owed assets.	0.0093	No	No
Workers arrived Thursday to the plant where the only person who remained on duty in the place who has not claimed his salary joined the protest.	0.0088	No	No

A useful application of identifying the key sentences is text summarization and visualization. Our model can assist a human analyst in quickly identifying the key information about an event without reading an entire document. Case Study 1 shows a demonstration of this practical application where key sentences within a news article are highlighted. From the highlighted sentences, we can easily find key information such as the which entity (*who*) against which entity, the details and reason behind the protest (*what, why*) and the location and time of the protest if available (*where, when*).

Table 4.5 shows the set of top positive sentences ordered by probability scores, as well as the set of negative sentences. Different event roles are also being highlighted with different

Workers System Veracruz Water and Sanitation (SAS) protested to demand pay bonuses and savings.

The more than thousand workers left the vicinity of the offices of SAS to address the Cathedral of Our Lady of the Assumption.

Some of the protesters walked without shoes and with the image of San Jose Obrero shoulder, whom they called the miracle of payment of bonus, salary, savings and benefits.

On Thursday, the workers and their wives staged a sit where they protested against the Municipal Palace cacerolazos of Veracruz.

The protest ended with a Mass at the Cathedral of Veracruz, where there was barely a capacity to accommodate more than a thousand workers participating in the walk of more than five kilometers performed with the holy shoulder.

Angelica Navarrete, general secretary of the Union of SAS, insisted on Tuesday that if they do not receive what they owe, they will strike.

During the march, at the height of Zamora Park, a passenger bus of the coastline they were pounced on protesters, upset because he wanted to spend and the march went through, but no injuries.

According to the protesters, the SAS, owed to workers 85 thousand 300 million pesos.

.....

Case Study 1: Key sentences are highlighted within a protest news article.

colors in the text. [2] We report common patterns among the positive sentences. For instance, most of them contain the location information such as *in Plaza Juarez, in the Eusebio Ayala Avenue*. Another common pattern is that the indicative sentences often contain some purpose-indicating words such as *demand, against*. From analyzing the negative sentences, we find that they may include some protest related words such as $protest, protestor, parade$, but are assigned lower scores because of the lack of protest action pattern and contextual information.

Further, in the last two columns of Table 4.5, we show whether the keywords and start/end methods also select our high ranked sentences as key sentences. We find that the keywords method has a high overlap with our method for the positive sentences. However it also introduces false positives as shown for the negative sentences.

Event Type and Population-Specific Tokens

For every protest article, the GSR provides a specific classification as it relates to the event "population" and "type". Representing the protest articles by the identified key sentences we extract the most frequent words within these sentences and report them in Figure 4.5 in descending order of the normalized frequency score. Specifically, for each class c_p and c_e in event population and event type, we assign a score to each word w to evaluate it's contribution to a given class. The score function is a normalized word frequency given by:

$$\text{Score}_c(w) = f_{c,w} \log \frac{N}{n_w}, \tag{4.4}$$

where, $c \in \{c_p, c_e\}$, $f_{c,w}$ is the frequency of token w for class c, n_w is number of documents containing w. N is the total set of articles. From Figure 4.5, we see that many of these terms are recognizable as terms about Business, Media and Education (event population) and Housing and Economic (event type). For instance, terms such as "sellers" and "commercial" have been chosen as top words in the key sentences in business articles. "Students", "education" and "teachers" are selected with higher weights in news articles in education category although some neutral words such as "national" are also identified.

[2] The text examples listed in this section are translated from Spanish to English using Google Translate Tool.

EventPopluation					EventType				
Business	**Media**	**Medical**	**Legal**	**Education**	**Housing**	**Energy**	**Economic**	**Employment**	**Government**
sellers	communicators	health	grant	students	housing	water	producers	worker	national
commercial	journalists	medical	congress	education	neighborhood	energy	mobilization	official	march
drivers	express	hospital	judges	national	service	company	route	drivers	government
strike	agreement	unemployment	specialties	government	terms	sector	budget	payment	demand
transport	exhibited	doctor	reprogramming	teachers	family	neighbors	carriers	wages	square
measure	profession	nursing	budget	college	group	lack	association	unemployment	city
carriers	legislation	clinics	explanation	professor	transfers	supply	ministry	guild	front
public	guards	patients	deny	faculty	place	population	cooperators	employee	hours
municipal	intervened	welfare	approve	school	mutual	authority	peasants	company	demonstration
strength	collaboration	power	exist	dean	bill	organization	PLRA	job	students

Figure 4.5: Top scored terms in different categories of event populations and event types. All the articles are represented by the MI-CNN model selected key sentences.

Event Encoding

As a downstream application, we explored the capability of encoding (extracting event information) events from the identified key sentences. Since, the event encoding task is not the main focus of our work, we try previously developed open information extraction tools for this purpose.

We use ExtrHech [131], a state-of-the-art open information extraction tool. ExtrHech is a Spanish Open IE tool based on syntactic constraints over parts-of-speech (POS) tags within sentences. It takes sentences as input and outputs the relations in the form of tuples (argument 1; relation; argument 2). Table 4.6 shows a list of events extracted by ExtrHech. We notice that ExtrHech is good at capturing the event population and action information, but not good for the "event type" information. The reason might be that the syntactic rules in ExtrHech are more suitable for capturing the pattern (Subject, Predicate, Object). For instance, ExtrHech captured entity words such as "campus" (indicating education), "pension" , "producers" (indicating business), "mayor", "gendarmes" (indicating Legal).

4.5 Related Work

4.5.1 Event Extraction

Event detection/extraction with online open source datasets has been a large and active area of research in the past decades. In political science field, there have been several systems such as GDELT [67], ICEWS [89], and EL:DIABLO [103] working on extracting political events

Table 4.6: List of events extracted using ExtrHech

Argument 1	**Relation**	**Argument 2**
the retired	require pension	Social Security Institute Servers
the protesters	complain	Guerrero campus
the manifestation	cause trouble	passangers
the district	organize	carnival
the protesters	are required	councilors
Antorcha Campesina organization	agglutinated	the capital
the situation	annoy	producers
the mayor	demand expulsion	colonists
gendarmes	ensure	conflicts

from online media. Supervised, unsupervised, and distant supervision learning techniques have been developed to tackle different domains and challenges.

Supervised learning approaches often focus on handcrafted ontologies and heavily rely on manually labeled training datasets at the sentence, phrase, and token levels. Chen and Li *et al.* [21, 70] utilize the annotated arguments and specific keyword triggers in text to develop an extractor. Leveraging social network datasets, Social Event Radar [48] is a service platform that provides alerts for any merchandise flaws, food-safety related issues, unexpected eruption of diseases, or campaign issues towards the government through keyword expansion. Social streams such as Twitter [120, 99] have been used for event records extraction and event detection. Event structure in open domains are mostly complex and nested. Supervised event extraction [78],[64] has been studied by analyzing the event-argument relations and discourse aspects of event interactions with each other. Even though, these methods often achieve high precision and recall, they do not scale to large datasets due to the limited availability of low level labeled data. Different from these approaches, our method utilizes the multi-instance formulation to propagate the labels from article level to sentence and phrase level. The proposed method is suitable because training data is easily available at the document level rather than per-sentence level.

In the unsupervised setting, approaches have been developed [74, 87] that model the underlying structure by jointly modeling the role of multiple entities within events or modeling

an event with past reference events and context. Approaches [18, 84] extract the event without templates based on probabilistic graphical models. The advantage of unsupervised approaches is that they don't require any labeled data and might be able to use the large quantities of unlabeled data, available online. The disadvantage of unsupervised methods is that they might suffer due to noisy information and concept drift.

Between supervised and unsupervised approach, distant supervision methods try to mitigate their disadvantages and often utilize the public knowledge base to generate training samples. Mintz et al. [82] use Freebase relations and find sentences which contain entities appearing in these relations. From these sentences, they extract text features and train a classifier for relation classification.

4.5.2 Multiple Instance Learning

Multiple Instance Learning (MIL) [29] is developed for classifying groups of instances called "bags". In standard MIL formulation, individual instance level labels are not available and labels are provided only at the group/bag level. Each *bag* is labeled positive if it contained at least one positive instance and negative otherwise. This MIL formulation makes strong assumptions regarding the relationship between the bag and instance-level labels. There are approaches that extend Support Vector Machines (SVM) for the MIL problem [9, 41] which include: (i) modifying the maximum margin formulation to discriminate between bags rather than individual instances and (ii) developing kernel functions that operate directly on bags (MI-SVM, evaluated in this work). Specifically, the generalized MIL [117] assumes the presence of multiple concepts and a bag is classified as positive if there exists instances from every concept. Relevant to our work, besides predicting bag labels, Liu *et al.* [73] seek to identify the key instances within the positively-labeled bags using nearest neighbor techniques. The recent work of [62] focuses on instance-level predictions from group level labels (GICF) and allows for the application of general aggregation functions while inferring the sentiment associated with sentences within reviews. Similar to our idea, Hoffmann and Surdeanu *et al.* [47, 110] utilize external knowledge base to extract relation from text in MIL framework. Different from traditional distant supervision, they assume that if two entities participate in a relation, then at least one sentence that contains these two entities might express that relation. Different from these work, we don't have an external source to

determine the involved entities in the events.

4.5.3 Convolutional Neural Networks

Convolutional Neural Networks (CNN) have found success in several natural language processing (NLP) applications such as part-of-speech tagging, chunking, named entity recognition, and semantic role labeling [24]. Kim [59] applies CNN to text classification for sentiment analysis. Kalchbenner *et al.* [57] propose a CNN approach to model the sentence vector based on dynamic k-max pooling and folding operation. Shen *et al.* [105] propose a latent semantic model based on CNN to learn a distributed representation for search queries. In our work, we use CNN to learn sentence representations by combining both local and global information and couple this representation within a relaxed MIL formulation.

4.6 Summary

We propose a novel method to extract event-related sentences from news articles without explicitly provided sentence-level labels for training. Our approach integrates a convolution neural network model into the multi-instance learning framework. The CNN model provides a distributed sentence representation which combines local and global information to relax the independence assumptions of standard MIL formulations. We perform a comprehensive set of experiments to demonstrate the effectiveness of our proposed model in terms of classifying a news document as a protest or not and extracting the indicative sentences from the article. Using the identified sentences to represent a document, we show strong classification results in comparison to baselines without use of expert-defined dictionaries or features. The strengths of our proposed model is highlighted by integrating with visualization and summarization applications as well as detection of finer patterns that are associated with an event type and population.

Chapter 5

Multi-Task Multi-Instance Recurrent Neural Network for Event Extraction

Along with the exploding number of digital data sources, the demand of making use of extracted information during decision making process becomes increasing intensive[11, 51, 88, 4]. In various applications such as algorithmic trading, media monitoring and risk analysis, extracting information has played a crucial part in the whole system. As a special form of Information Extraction, Event Extraction (EE) has attracted growing research interest in recent years. Li et.al. [70] proposed a successful joint model which is based on the structured perceptron algorithm with a rich set of global and local features to simultaneously predict the event trigger and arguments. Most traditional approaches usually rely on domain and language knowledge to design the features. This might limit the application of models developed in one domain to other domains. In recent years, deep learning approaches [21, 85, 114, 5] have been widely applied in the task of event extraction and achieved the state of art performance. One key benefit of deep learning is that the algorithms automatically learn the representation of input and avoid manually engineering and carefully crafting the features. However, most of these approaches require accurate annotation in entity mention level, which is complex and time-consuming to label. To address this challenge, we propose a distant supervision based approach which utilizes the existing event database to substitute the requirements of accurate labeling. We assign the label to the entity level rather than directly give each entity mention a label.

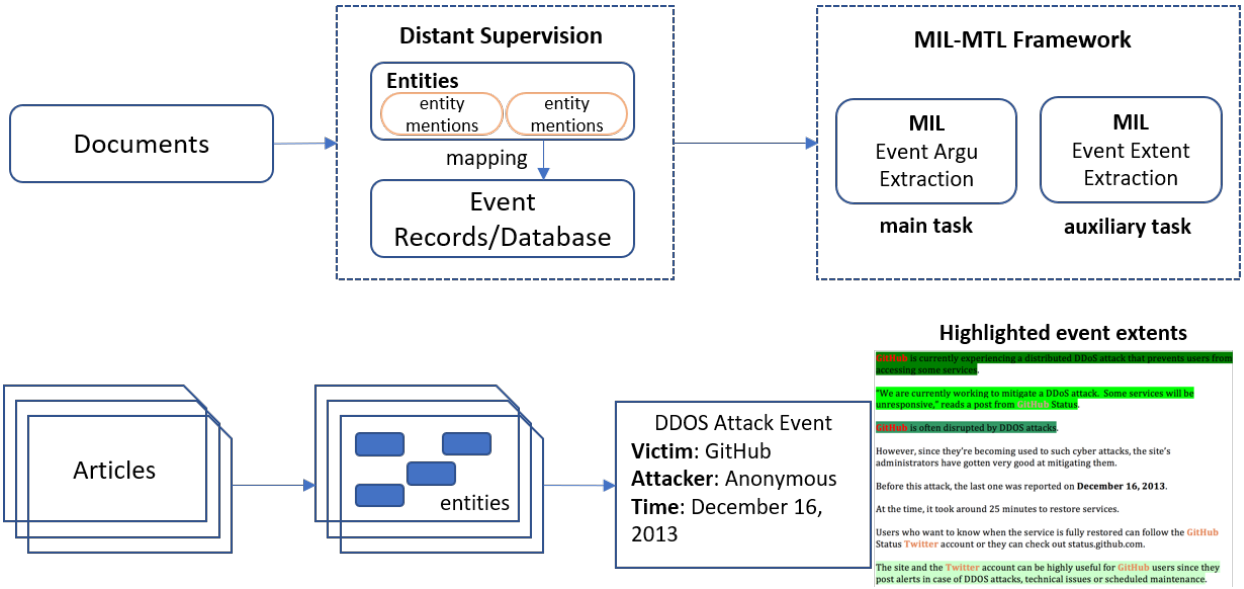

Figure 5.1: System Overview

It is common in event extraction task that the entity mentions belonging to the same entity might take different event roles in one article.

To motivate our approach, consider the following two sentences that refer to the entity **Github** but in different contexts.

S1: ***Github*** *has been hit by another distributed denial-of-service attack launched by the hacker group Anonymous on March 11 at 14:25 PTDT*

S2: *As most* ***Github*** *users will tell you, DDoS attacks against the service are highly common.*

In both sentences S1 and S2, **Github** is considered as a entity reference or mention and the objective of our work is to determine it's role/context. **Github** is the Target of a DDoS attack in S1. However, in S1 of the same article **Github** does not take any role in this event.

In this study, we view the classification of event arguments as a form of Multi-Instance Learning (MIL). It enables us to identify the entity mentions involved in the event without the phrase level labeling. A pertinent challenge in training machine learning models for domain specific event extraction is that the size of the training dataset for domain event is usually small (hundreds events). Inspired by the successful applications of Multi-Task learning (MTL) in natural language processing [23, 22, 97], we design a Multi-Task strategy

to alleviate the problem of limited training data. In addition to the main task of identifying event arguments, we design an auxiliary task of extracting event-related sentences (referred by *event extents*) from the document. The tasks of classifying event argument and identifying event extents are strongly correlated to each other. The event extents should contain at least one event argument. Moreover, the MIL strategy also works for the extraction of event extents from the article without sentence level labeling. Figure 5.1 demonstrates the overview of our proposed approach. Given a typical news article, we first identify all the entities; and then extract the event arguments as well as the key sentences (event extent) as a byproduct. As shown in Figure 5.1, the highlighted event extents can help users quickly locate the key information within an article of interest.

We propose a Multi-Instance, Multi-Task RNN (MIMTRNN) approach for event extraction with distant supervision. The distant supervision will reduce the effort and cost for event labeling. The MIL module will handle the noise labels from distant supervision. The MTL module benefits the framework from two ways. First of all, the MTL increases the number of samples that we are using to train the model. Secondly, Modeling the two related tasks together reduces the risk of over-fitting for each task and makes the learned representation more general. As a proof of concept, we validate our method on two real-world event datasets: cyber-attack and social unrest event set.

5.1 Notations and Problem Setting

5.1.1 Problem Statement

In this work, we aim to extract the interested event for specific domains, such as cyber-attack event. In the training phase, we are given a set of event records for a specific domain and corresponding news articles reporting the event. Each event record consists of a set of event arguments. Each entity might have multiple entity mentions in the article, while we do not have the ground truth label for each entity mention. We do not know which entity mention is really contributing to that event argument. In the test phase, the task is to assign the event roles to the entities if the article refers a specific interested event. Taking the Cyber Attack event extraction task as example, the arguments of a cyber-attack event consist of

Attacker, Target and Time. After training, the model will assign one of the four classes (Attacker, Target, Time, None) to each entity in the test article.

5.1.2 Notations

We follow the same concepts defined in the ACE (Automatic Content Extraction) event extraction task [30].

- **Entity:** An *Entity* is defined as an object or set of objects in the world. E.g., **Micorsoft** is a company entity and **Anonymous** is a organization entity.
- **Entity Mention:** Entity mention is a reference to the entity in the text. E.g., each appearance of *Micorsoft* in one article is a mention for **Microsoft** entity.
- **Event Extent:**: An Event extent is a sentence within which a specific event is described.
- **Event Argument:** Event argument is an entity mention, which will be a participant or an attribution in the event.

5.2 Model

We propose a Multi-Instance Multi-Task Learning model based on Recurrent Neural Network (MIMTRNN) for our task. We utilize the Recurrent Neural Network to construct the distributed representation of entity mentions and sentences. The Multi-Instance Learning module takes as input the instance representation and via backpropagation forces the RNN module to learn better representation. For each entity mention/sentence instance, the model estimates a probability distribution over the target classes; and then an aggregation function is applied on the instances to infer the probability distribution of the bag over classes. In addition to sharing the underlying token embeddings, the argument prediction module and event extent identification module are trained together to overcome the small size of training data and overcome over-fitting risk for each task. Figure 5.2 provides an overview of our proposed model. x_i to x_N are the input representations of the tokens in each sentence, and

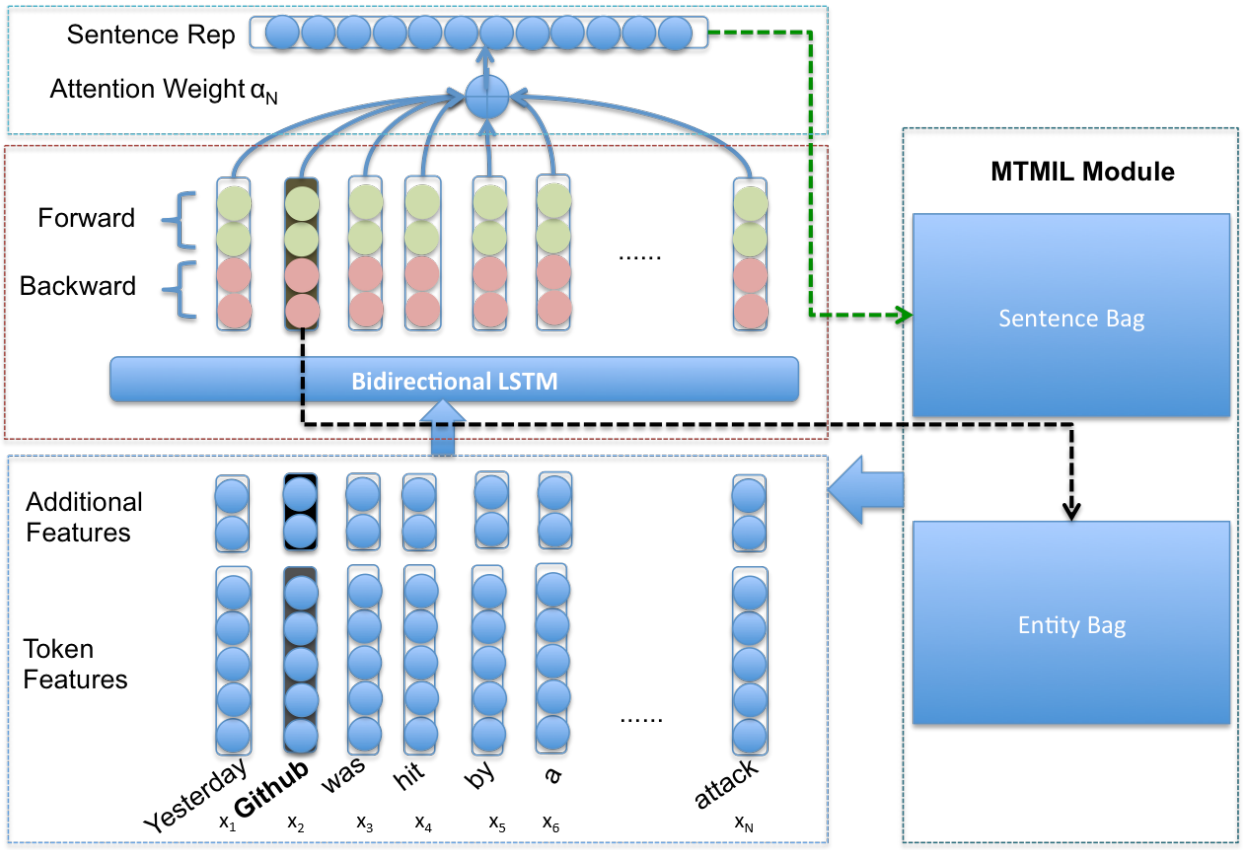

Figure 5.2: Multi-Task Multi Instance RNN Model Overview

the tokens with the black ground are the specific entity mentions. In addition to the word embedding features, other types of information, such as entity type can be easily integrated into this framework.

5.2.1 Encoding Entity Mention and Sentence

Context Word Window

The main input to the system is a one hot representation of the tokens and the entity mentions' positions in the sentence. By looking up the word embedding matrix, each token is transformed into a continuous space representation. Typically the word embedding matrix is pre-trained using large external corpus such as Wikipedia by shallow neural model. The learned word embedding has some interesting properties [81] and tends to cluster the words with similar semantic [24].

As suggested in [79], the context word window captures short-term temporal dependencies surrounding the word of interest. Before feeding the input into the Bidirectional LSTM layer, we replace each token in the sentence with a context word window. Given a window of size $2d+1$ and the dimension of the word embedding e, the context word window x_t is built by the concatenation of the previous d words followed by the current word and d following words.

$$x_t = [w_{t-d}...w_t...w_{t+d}] \in R^{(2d+1)\star e}$$

For the first and last word, we add a special token *padding* d times to the begin and end of the sentence. The following is an example of context window of size 3 for word **Anonymous** in the sentence S3.

S3: *The online hacktivist* ***Anonymous*** *has taken down the official website of* ***Fullerton police department*** *in retaliation for the arrest of protesters.*

$$x_{Anonymous} = [w_{hacktivist}, w_{\mathbf{Anonymous}}, w_{took}] \in R^{3e}$$

Bidirectional LSTM

The proposed model seeks to learn representations of both, the entity mention and sentence. We use RNN [91] to encode the entity mention and sentence into dense vectors. RNN has been widely used in different NLP tasks [79, 123, 85] to model sequential dependencies within text data.

For event extraction case, the label of entity mention is not only related to previous tokens in the sentence but also the following ones. For instance, in sentence S3 the tokens *has taken down* after ***Anonymous*** and the one *hactivist* before it are strong surrogates that suggest ***Anonymous*** is the attacker for the cyber attack incident. Similarly, assigning the label *Attacker* to ***Anonymous*** helps the system correctly classify ***Fullerton police department*** and vise versa. Follow this idea, we use bidirectional LSTM to encode sentence from both directions with different hidden layers. The final output $\bar{h_t}$ at time step t is the concatenation of forward output $\overrightarrow{h_t}$ and backward output $\overleftarrow{h_t}$.

$$\bar{h_t} = [\overrightarrow{h_t}, \overleftarrow{h_t}]$$

Representation for Entity Mention and Sentence

We treat each entity mention as a token, and the representation of the entity mention is the output of the Bidirectional LSTM at its position in the sentence. In the case of multi-tokens entity mention, we transform the entity mention into one new token by concatenating all the original tokens. For instance, the GPE entity mention “United States" is replaced by the new token United_States.

The most common way to build the sentence embedding in Bidirectional LSTM is to concatenate the representation of the last token in both forward and backward directions. The output of the last LSTM cell encodes everything we need to know about the sentence. In some cases, the key information indicating the occurrence of target event might be far away from the start and end of the sentence. For example, in the following sentence:

S4: *As a result of arrest and acquittal of police officers,* ***Anonymous*** *decided to take revenge in the form of* ***taking down*** *the official website* ***Fullerton police department****, which at*

the time of publishing this article was still down and was showing error.

The key phrase **taking down** indicating the cyber attack event is far from the begin and end of the sentence. In theory, LSTM can deal with arbitrary long sequences, but in practice, the long-range dependencies are still problematic. Alternatively, we utilize the attention mechanism to learn the sentence representation. With the attention mechanism, we no longer rely on the output of the last LSTM cell to include all the information of a sentence. Rather, we allow the model to put particular attentions on different parts of the sentence. As shown in Figure 5.2, the sentence representation r_s is a weighted combination of all the LSTM cell outputs.

$$r_s = \sum_{t=1}^{N} \alpha_t h_t \tag{5.1}$$

$$\alpha_t = \frac{exp(w_\alpha \cdot h_t)}{\sum_{j=1}^{N} exp(w_\alpha \cdot h_j)} \tag{5.2}$$

Here, w_α is the attention parameter which is a vector with the same length as h_t. α defines contribution that a token makes to the sentence embedding. Higher value of α implies that the sentence embedding pays more attention on that part.

5.2.2 Multi-Task Multi-Instance Learning

In the context of event extraction, it is common that not all the mentions of one entity in the text belong to the same event argument. For the argument entity, only few of mentions take the event role and the remaining take the role of "**None**". Similar to the case of entity mention, the sentences containing the argument entities may not be all event extents. Based on these ideas, it is natural and flexible to use Multi-Instance Learning paradigm to model the entities and sentences.

The main goal of this work is to identify the event and extract the event arguments from the text. We consider the task of classifying the entities into different argument roles as our main task. We design an auxiliary task of identifying event extent in addition to the main task.

We consider each entity as a bag and all its mentions in an article as instances. Similarly, in training phase, we consider all the sentences containing at least one event argument as a positive bag and the remain sentences constitute the negative bag in one article. As shown in the Figure 5.2, the main and auxiliary tasks share the word embedding layer and Bidirectional LSTM layer. The intuition behind this task pair is following. On one hand, the sentences which are event extents must include at least one event argument. On the other hand, the entity mentions which are event arguments must appear in the event extent.

Given a article with n entities and m sentences, the training set D is $[(E, L_E), (S, L_S)]$, where $E = \langle E_1, ..., E_n \rangle$ is a set of entity bags and $S = \langle S_1, S_2 \rangle$ is a set of sentence bags. L_E, L_S are sets of labels for entity and sentence bags, respectively. The ith training entity is represented by $\langle E_i, \ell_{Ei} \rangle$, where the entity bag E_i is associated with the label ℓ_{Ei}. $E_i = \langle e_{i1}, ..., e_{in_i} \rangle$ is a collection of n_i entity mentions described by r_e the output of one Bidirectional LSTM cell. Each sentence bag $S_i = \langle s_{i1}, ..., s_{in_i} \rangle$ is a set of sentences and each sentence is represented by r_s the weighted combination of the LSTM cell outputs with attention mechanism.

Classical MIL algorithms [7] follow two types of approaches. An instance-based model infers the instance level labels first and then derive the bag level labels from instances' labels. On the other hand, a bag based model attempts to directly obtain the bag level label based on bag discrimination information. In this work, we take the instance based paradigm to infer the bag level labels.

Given the representation r_e of the entity mention, we compute the probability p_e of the entity mention over argument labels by a softmax function.

$$p_e^k = softmax(r_e) = \frac{exp(\theta_k \cdot r_e + b_k)}{\sum_{j=1}^{K} exp(\theta_j \cdot r_e + b_j)} \tag{5.3}$$

The probability p_s of the sentence mentioning an event is calculated by a sigmoid function given the sentence embedding r_s.

$$p_s = \sigma(\beta \cdot r_s + b_s) = \frac{exp(\beta \cdot r_s + b_s)}{1 + exp(\beta \cdot r_s + b_s)} \tag{5.4}$$

Here, θ, β and b are model parameters.

We consider the bag level probability distribution as the aggregation output of the instance level probability distribution and apply a Noise-OR mechanism on the bag level aggregate function. The Noise-OR strategy assumes that the positive bag contains at least one positive

instance, while all the instances in the negative bag are negative. However, it is not known that in positive bag which instance is positive and whether there are more than one positive examples in the bag. The probability of the bag being positive is decided by the most positive instance. On the other hand, the probability of the bag being negative is determined by the least negative instance. For sentence bag (binary case), the probability of the bag being positive (include event extent) is calculated by:

$$p(\ell_{Si} = 1) = \max_{s \in S_i} p(\ell_s = 1) \tag{5.5}$$

For the multi-classes case with $|L|$ types of labels, we assume that the first $|L|-1$ classes are positive and the last one is negative class. In the context of cyber attack event extraction, we consider the argument classes (**Attacker, Target** and **Time**) are positive and **None** as negative. We compute the unnormalized bag probability for class i by:

$$\tilde{p}(\ell_E = i) = \begin{cases} \max_{e \in E} p(\ell_e = i) & \text{if } i \in [1, ..., |L| - 1] \\ 1 - \max_{j \neq i} \tilde{p}(\ell_E = j) & \text{Otherwise} \end{cases} \tag{5.6}$$

Then, the bag probability over classes is estimated by:

$$p(\ell_E = i) = \frac{\tilde{p}(\ell_E = i)}{\sum_j^{|L|} \tilde{p}(\ell_E = j)} \tag{5.7}$$

We use E for the entity bags, S for the sentence bags and y for the labels in one article. With the auxiliary task, we define the loss function for one article as below:

$$L(E, S, y; \theta) = \lambda \frac{1}{N_E} \sum_{e \in E} -y_e \cdot log(p(e; \theta)) + \frac{(1-\lambda)}{2} \sum_{s \in S} -y_s \cdot log(p(s; \theta)) \tag{5.8}$$

where θ represents all the model parameters, N_E is the number of entity bags in that article and λ is the loss weight with the value between 0 and 1. The first part of loss function is the cross-entropy loss for the main task based on the entity labels. The second part is the cross-entropy loss for auxiliary task based on the sentence bag labels.

The training objective is optimized using mini-batch stochastic gradient descent (SGD) and we use one article as a batch during the training. We also apply Dropout on model layers

to further reduce the overfitting. There are several common used optimization methods when training with SGD, including AdaGrad [33], AdaDelta [126] and RMSprop [111]. Empirically, RMSprop works the best on our dataset, so we only report the results with RMSprop.

5.3 Experiments

5.3.1 Datasets

We evaluate the performance of our proposed approach on two real-world datasets: (i) Cyber Attack Event Set and (ii) Social Unrest Event dataset.

Cyber Attack Events We collect a set of distributed denial of service (DDoS) events from the website *http://www.hackmageddon.com/*. Each DDoS event consists of three event related fields (Attacker, Target, Time) and the news articles referring the event. For the training set, we first extract all the entities with types: (i) Person, (ii) Organization, (iii) Geopolitical Entity (GPE) or (iv) Time in the article and then map the event arguments to these entities. Finally, the unmatched entities are assigned the *None* label.

Social Unrest Events In our experiments, we use a manually labeled civil unrest dataset from news article. Each protest event provides the population, protest type and location of the event, as well as the article reporting the event. Each annotated sample is checked by three human analysts and the labels are confirmed if two of them agree on the assignment. In this task, we will predict whether the location mentioned in the news articles is involved in a protest or not.

Table 5.1 shows two examples for the events used in our experiments. We use 5-fold cross validation for evaluation and report the mean precision, recall and F1 score along with standard deviation across five folds. Table 5.2 shows the summary statistics for one of the 5 folds for these two datasets.

Table 5.1: Event record examples for Cyber Attack and Social Unrest. (Note: the event sentences are not given in the dataset, we add the sentences here to give the readers more context information about the event.)

Cyber Attack		**Civil Unrest**	
Type	DDoS	**Type**	Government Policy
Attacker	Anonymous	**Location**	London
Victim	Angola		
Time	Sunday		
Sentence	Anonymous take down Angola government website on Sunday.	**Sentence**	Around 20,000 pro-refugee demonstrators took to London streets on Saturday.

5.3.2 Baseline Methods

In this section, we discuss a number of strong baseline approaches which are evaluated in our experiments.

Baseline 1: SVM with Text-based Features

Linear Support Vector Machine (SVM) [53], which is widely regarded as one of the best text classification algorithms. In addition to the entity token itself, we design four other types of features to represent the entity mentions.

Token Context Window The surrounding tokens of the entity mention in the sentence with window size 3.

Dependency Parsing Tree The neighbor of the entity mention in the dependency parsing tree.

Entity Type The type of the entity, such as Organization, Time and so on.

Title information Whether the entity appears in the title or not.

Table 5.2: Summary of Cyber Attack and Civil Unrest dataset

	Events	Sens	Entities	Entity Ments
Cyber Attack				
Train	100	1636	1206	1772
Valid	20	299	288	437
Test	35	566	478	757
Civil Unrest				
Train	612	5658	3238	10064
Valid	203	1884	1084	3326
Test	204	2011	1089	3235

Baseline 2: Hierarchical RNN

Lin et al. [71] proposed a Hierarchical Recurrent Neural Network (HRNN) approach to model the document. Inspired by this idea, we use the HRNN to build the representation for an entity. We represent each entity mention by concatenating the token and its relative position to the entity mention in the sentence. After adding the relative position to the tokens, each entity mention could be considered as a different sentence. Then similar to the application of HRNN on the document, we apply one LSTM layer on entity mention tokens to get the entity mention embedding and apply another LSTM layer on mention embedding to get entity embedding. Finally, a softmax layer is applied on the entity embedding to compute the probability distribution over event argument labels.

Baseline 3: Hierarchical CNN

Denil el at. [27] proposed a hierarchical CNN document model to build the document representation and extracted the salient sentences. The approach involved use of two CNN layers to encode sentence and document. Both RNN and CNN are frequently used in text related tasks. RNN is a more 'natural' approach as it is able to capture the sequential dependence prevalent within natural language. Compared with RNN, CNN models are

Table 5.3: Event Extraction performance. Comparison based on micro-average Precision, Recall and F-1 Score w.r.t to baseline methods.

	Cyber Attack			**Civil Unrest**		
	Precision(Std.)	Recall(Std.)	F1(Std.)	Precision(Std.)	Recall(Std.)	F1(Std.)
SVM	0.77(0.037)	0.79(0.028)	0.78(0.031)	0.84(0.012)	0.85(0.29)	0.84(0.031)
H-RNN	0.76(0.020)	0.71 (0.041)	0.73(0.018)	0.85(0.031)	0.86(0.076)	0.85(0.022)
H-CNN	0.75(0.031)	0.74(0.018)	0.74(0.025)	0.86(0.007)	0.86(0.066)	0.86(0.013)
MI-RNN	0.71(0.034)	0.69(0.008)	0.70(0.007)	0.85(0.038)	0.85(0.042)	0.85(0.056)
MI-CNN	0.78(0.070)	0.77(0.038)	0.77(0.040)	0.84(0.068)	0.85(0.036)	0.86(0.035)
MIMT-RNN	**0.83(0.036)**	**0.81(0.090**	**0.82(0.013)**	**0.89(0.033)**	**0.90(0.088)**	**0.89(0.045)**

more often than not used in the cases where the feature detection is more important. Based on this idea, it's natural to replace the RNN layer in the baseline 2 CNN layer to encode the entity mention and entity bag.

Baseline 4: Basic MI-RNN

In the context of Multi-Instance Learning framework, the first three baseline methods all belong to the category of bag-based approaches. In more detail, they are mapping-based classifiers [46], which transform each bag into a single-instance representation such that any single-instance classifiers can be trained to predict bag labels. We use another MIL paradigm instance-based approach to infer the bag level probability over labels. We make use of the first RNN layer to encode the entity mention representation. Rather than continue to build the entity representation, we directly determine the instance labels and use them to derive bag label with an aggregation rule and refer to this approach by Basic MI-RNN

Baseline 5: Basic MI-CNN

Similar to the basic MI-RNN approach, the basic MI-CNN model also takes the instance based paradigm to infer bag level label. We use the same first CNN layer of the Hierarchical CNN model to build entity mention representation rather than RNN.

5.3.3 Results and Discussion

Table 5.3 shows micro-average of precision, recall, and f1 for our approach MIMTRNN and other comparative methods on cyber-attack and civil unrest dataset. The results show that our proposed MIMTRNN approach significantly outperforms other comparative methods on both datasets. As expected, the SVM baseline method also demonstrates a strong result on both dataset. In the cyber-attack dataset, SVM approach outperforms other deep learning baseline approaches and achieves the second best performance. The size of cyber-attack data set is relative small and the simple SVM model with hand-crafted features generalizes better than the complex deep models without proper regularization. For the civil unrest dataset, along with the increasing data size, the deep learning models start to achieve close or even better performance than the SVM model. However, the SVM approach requires substantial effort on feature engineering. This kind of requirement also makes the same model hard to apply on other domains. More importantly, the hand-crafted context features might fail to capture the long distance dependencies, which are important in event extraction.

The representation learning module (RNN) in our approach is able to learn the context feature automatically. As shown in Table 5.3, the MI-CNN approach is close in performance to the SVM approach for the Cyber Attack dataset and H-CNN approach has better performance than SVM for the Civil Unrest dataset. The convolution neural network learns meaningful embedding for the entity mention. The overall CNN baseline models performs better than the RNN baseline models under the same setting, and the MI-RNN model achieves the poorest result among all the approaches.

The proposed MIMTRNN model, instead of modeling each entity mention independently, encodes all the entity mentions in one sentence together. Furthermore, the Bidirectional LSTM is able to capture both past and future information. Compared with HCNN model, the MI-CNN model attains a significant performance improvement by applying the noise-or mechanism to infer bag label from instance labels on the Cyber Attack dataset. This improvement confirms our MIL assumptions that not all the mentions of one entity share the same label, and the label of the bag is decided by the most positive instance in our case.

In addition to the model structure, we run a set of extensive experiments to analyze the impact of different strategies for training word representations, use of additional features and auxiliary tasks.

Table 5.4: Performance of different word embedding strategies

		Precision	Recall	F1	Precision	Recall	F1
		Random			Pretrained		
Cyber-Attack	Dynamic	0.82(0.019)	0.81(0.034)	0.81(0.028)	0.83(0.036)	0.81(0.090)	**0.82(0.013)**
	Static	0.69(0.023)	0.70(0.011)	0.70(0.008)	0.72(0.017)	0.74(0.023)	0.73(0.030)
Civil-Unrest	Dynamic	0.86(0.101)	0.90(0.046)	0.87(0.029)	0.89(0.033)	0.90(0.088)	**0.89(0.045)**
	Static	0.82(0.058)	0.85(0.011)	0.83(0.047)	0.85(0.098)	0.88(0.043)	0.86(0.033)

Impact of word representations

We investigate the impact of different strategies for training and using word embeddings. We focus on the two types of strategies: using pre-trained word embedding (Pre-trained) or not (Random) and updating the word embeddings during training (Dynamic) or not (Static). There are several popular pre-trained word embeddings [92, 80] from large corpora such as Wikipedia and Google News. Specifically, we report the result for 50 dimension Glove word vectors pre-trained on Wikipedia and Gigaword. Table 5.4 shows the performance of the four combinations on the two type of strategies. The first observation is that updating the word embedding parameters during the training process has a great impact on the performance for the cyber-attack and civil unrest datasets. Given the pre-trained word embedding, the model which updates the word embedding during training process achieves 9% and 3% f1-score improvement for cyber-attack and civil unrest dataset in comparison to the same model structure without updating word embeddings, respectively. Updating word embedding in our framework benefits further in the case where the input word embeddings are randomly initialized. The second finding is that the Random initialization of word embedding has a comparable performance to the Pre-trained word embedding. One possible explanation might be that the event arguments might have a strong correlation to a small set of frequent keywords which can be learned well from the training corpus itself.

Impact of Additional Features

Although the representation learning module is able to automatically learn features for entity mentions, the model benefits from the input of domain knowledge. We will demonstrate later that it is easy to integrate additional features into our framework. Initially, our model only

Table 5.5: Performance Comparison with and without additional features (Cyber Attack)

	Token Feature			Token + Entity Type Feature		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Attacker	0.44(0.004)	0.63(0.016)	0.52(0.020)	0.45(0.021)	0.53(0.011)	0.49(0.020)
Target	0.70(0.023)	0.56(0.019)	0.62(0.008)	0.64(0.022)	0.73(0.061)	**0.68(0.081)**
Time	0.73(0.031)	0.49(0.019)	0.58(0.028)	0.69(0.029)	0.85(0.043)	**0.76(0.022)**
None	0.85(0.008)	0.90(0.010)	0.88(0.021)	0.91(0.008)	0.85(0.003)	0.88(0.012)
Avg / Total	0.80(0.023)	0.80(0.018)	0.80(0.021)	0.83(0.036)	0.81(0.090)	**0.82(0.013)**
	(a)			**(b)**		

incorporates the context window of the entity mention as input. Table 5.5 (a) shows the model performance for all event roles on cyber-attack dataset. Without using any other features except the context window, our model obtains a performance 0.80 F1-Score which outperforms all other baseline methods. Adding entity type feature for the cyber-attack event extraction task leads to a further improvement.

Table 5.6 shows the prediction errors for each event role for the cyber-attack dataset. As expected, the most common type of errors is that the event roles are falsely predicted as None and vice versa. There are several **Time** roles that are predicted as **Target**. This type of error can be avoided by introducing the entity type features, since the **Time** role must be the *time expression* entity. The **Attacker** and **Target** are usually the entities with types as *Organization*, *GPE* and *Person*.

There are two ways to integrate the entity type feature into our framework. (i) We use the one-hot representation for each entity type and concatenate it to the end of each instance representation in the bag. (ii) We define an entity type embedding which maps each entity type into a dense vector; and the type embedding is updated during the training process. We use the first approach to define entity type feature in the experiments to reduce the model complexity due to the small training size. Table 5.5 (b) shows that the F1-Score increases 18% for **Time**, 6% for **Target**, and finally 2% for the overall performance after introducing the entity type feature.

Table 5.6: Prediction Error in cyber-attack event without entity type feature

	False Positive		**False Negative**	
Attacker	Target	1	Target	3
	None	7	None	6
Target	Attacker	3	Attacker	1
	Time	3	None	27
	None	8		
Time	None	7	Target	3
			None	22
None	Attacker	6	Attacker	7
	Target	27	Target	18
	Time	22	Time	7

Table 5.7: Performance Comparison models with and without auxiliary task

	Without Auxiliary Task			With Auxiliary Task		
Cyber-Attack	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Attacker	0.50(0.019)	0.32(0.014)	0.39(0.017)	0.45(0.021)	0.53(0.011)	**0.49(0.020)**
Target	0.65(0.019)	0.71(0.052)	0.68(0.042)	0.64(0.022)	0.73(0.061)	0.68(0.081)
Time	0.54(0.014)	0.49(0.024)	0.51(0.019)	0.69(0.029)	0.85(0.043)	**0.76(0.022)**
None	0.87(0.007)	0.88(0.019)	0.87(0.008)	0.91(0.008)	0.85(0.003)	**0.88(0.012)**
Avg / Total	0.79(0.043)	0.80(0.038)	0.79(0.029)	0.83(0.036)	0.81(0.090)	**0.82(0.013)**
	(a)			**(b)**		
Civil Unrest	0.84(0.029)	0.87(0.049)	0.85(0.018)	0.89(0.033)	0.90(0.088)	**0.89(0.45)**

	Key Sentences Extracted From Test Articles	
Article No	**With Auxilary Task**	**Without Auxilary Task**
1	[Anonymous Takes Down 20 Thai Prison Websites] [Anonymous' crusade against the Thai justice system continues today, with the group bringing down 20 websites belonging to their prison system.] [Members of the Blink Hacker Group, who affiliate with the Anonymous movement, have taken credit for the attacks, which started last night]	[Previously, the same hacking crew also launched DDoS attacks against 300 Thai court websites.] [Members of the Blink Hacker Group, who affiliate with the Anonymous movement, have taken credit for the attacks, which started last night]
2	[HSBC has been battling an apparent Distributed Denial of Service (DDoS) attack on its online banking system for the past few hours] [HSBC blamed the outage on a DDoS attack, and attempted to spin the whole thing as a success story to mainstream news outlets] [HSBC internet banking came under a denial of service attack this morning]	[HSBC has been battling an apparent Distributed Denial of Service (DDoS) attack on its online banking system for the past few hours] [HSBC internet banking came under a denial of service attack this morning]
3	[On 30th January 2016 someone started carrying a series of powerful Distributed Denial-of-Service attack (DDoS) on Pastebin.com forcing the platform to go offline.] [Pastebin.com faced heavy DDoS attacks which forced the platform to go offline.]	[Hackers Target Pastebin.com with Powerful DDoS Attack] [On 30th January 2016 someone started carrying a series of powerful Distributed Denial-of-Service attack (DDoS) on Pastebin.com forcing the platform to go offline.]
4	[The Swiss Federal Railways website was hard to access on Monday afternoon for about an hour and in the evening for around one and a half hours due to a DDoS attack] [A group called NSHC claims responsibility for SVP hack and SBB DDoS assault.] [On Wednesday we were made aware that there was an attack.]	[A group called NSHC claims responsibility for SVP hack an d SBB DDoS assault.] [Swiss Federal Railways (SBB) and a number of retailers, including electronic retailer InterDiscount, were hit by Distributed Denial of Service (DDoS) assaults]
5	[Blighty's government-funded educational network Janet has once again been hit by a cyber attack, with a fresh wave of DDoS attacks launched against the network this morning.] [The issue first began on Friday 15 April, with the body reporting it had been hit by a DDoS attack]	[Blighty's government-funded educational network Janet has once again been hit by a cyber attack, with a fresh wave of DDoS attacks launched against the network this morning.]

Figure 5.3: Top three key sentences extracted by models with and without auxiliary task.

Impact of auxiliary task

To reduce the risk of over-fitting due to the limited training data, we design an auxiliary task of identifying the event extent along with the main task of event extraction. We systematically investigate the impact of the auxiliary task by comparing the performance between enabling and disabling the Multi-Task module. Table 5.7 shows the performance comparisons between two version of models trained with and without auxiliary tasks. For the cyber-attack task, the f1-score increases by 10% for Attacker, 25% for Time and 3% for overall performance. For the civil unrest experiment, the model improves around 4% on f1-score due to the auxiliary task.

To further analyze the affect of the auxiliary task, we compare two sets of top key sentences from models with and without auxiliary task for the cyber-attack dataset. The sentences

are ranked by the probability of the entity mention being an event argument. Figure 5.3 lists the set of top key sentences from five randomly chosen articles in the test set. First of all, both models seem to learn patterns of the event-related sentence that contain keywords like DDoS and attack. Moreover, the model with auxiliary task extracts more key sentences than the one without auxiliary task. The multi-task learning framework assists in learning common patterns among different event sentences. This could be specially useful for the case that the event sentence only contains reference of one single entity mention.

A Case Study

GitHub is currently experiencing a distributed DDoS attack that prevents users from accessing some services.

"We are currently working to mitigate a DDoS attack. Some services will be unresponsive," reads a post from GitHub Status.

GitHub is often disrupted by DDOS attacks.

However, since they're becoming used to such cyber attacks, the site's administrators have gotten very good at mitigating them.

Before this attack, the last one was reported on **December 16, 2013**.

At the time, it took around 25 minutes to restore services.

Users who want to know when the service is fully restored can follow the GitHub Status Twitter account or they can check out status.github.com.

The site and the Twitter account can be highly useful for GitHub users since they post alerts in case of DDOS attacks, technical issues or scheduled maintenance.

Figure 5.4: A case study for the extracted cyber attack event

The output of our MIMTRNN model can be used for event Summarization and visualization. The highlight sentences are the event extents extracted by our model and font colors are used to represent event roles. The scale of the color reflects the confidence of the prediction. As shown in figure 5.4, we use green color to highlight the event extents and the red font

color for **Target**. The first and third sentences have high confidence of mentioning a DDoS incident, while the last sentence gets the low confidence to be event extent even it contains the cyber attack related keywords. The red bold **GitHub** in the first and third sentences clearly indicates that it is the **Target** of a cyber attack event with high confidence.

5.4 Related Work

Distant Supervision for Event Extraction There has been considerable interest in extracting the event information from news and social media. However, usually these prior approaches use a supervised learning framework [21], or take a very coarse representation of event at the level of a sentence or tweet [93]. One of the limitations of supervised framework for event extraction is that it needs manual labeling of large number of instances that involve annotation of sentences within a text. Consequently, there has been increasing interest in distant supervision [25, 109], which allows for augmentation of existing events/knowledge bases from unlabeled text. Our work is related to [58], which uses the distant supervision to extract information about airplane crash events from news text. In [58], the argument labels are directly assigned to the entity mention level. However, in this work we do not assume the knowledge of labels at the entity mention level. Instead, we utilize the multi-instance learning framework to infer the event arguments.

Deep Learning in Event Extraction Inspired by the success of deep learning for image representation [63] and natural language processing [130, 122, 54], there has been increasing interest in applying deep learning to the task of event extraction [42]. Chen et al. [21] first utilize the pre-trained word embedding to capture lexical level features and apply a dynamic multi-pooling convolution neural network to capture sentence level features. Finally, a softmax operation is applied to classify the event trigger and arguments. Rather than classify each argument independently in [21], our approach makes use of RNN to encode entity mentions and implicitly models the correlation between labels of entity mentions in one sentence. Nguyen et al. [85] proposed a joint model which identifies the event trigger and arguments simultaneously. The joint model utilizes the Bidirectional LSTM to build the representation for entity mentions. A memory matrix is used to capture the dependencies between argument roles and trigger sub-types. In our work, instead of explicitly storing

the previous predictions by the memory matrix, we rely on Bidirectional LSTM to capture the dependencies between entity mentions. Moreover, we utilize the Multi-Task Learning module to further regularize the learning process.

Multi-Instance Learning in information extraction To avoid the laborious work of manually labeling the dataset at a fine level, distant supervision paradigm [82] has been widely applied in the domain of information extraction, especially for relation extraction. These methods are based on the assumption that if a relation exists in a pair of entities in knowledge bases, then every document containing the mention of the entity pair expresses the same relation. To alleviate the strong assumption of distant supervision, Riedel et al. [98] proposed a Multi-Instance Learning based approach to model the problem. In [98], each entity pair is considered as a bag which consists of all the sentences that contain the mention of the entity pair. Due to the strong power of representation of the neural network, it is natural to integrate deep learning model within multi-instance learning framework. Zeng et al. [127] proposed a piecewise convolution neural network (PCNN) model for relation extraction. Our work is very close to [72, 127], while to the best of our knowledge, our work is the first attempt to jointly applying Multi-Task, Multi-Instance Learning and Deep Learning on the task of event extraction.

5.5 Summary

We propose a novel approach to extract event from news article using distant supervision. Our approach integrates a Bidirectional LSTM module, Multi-Task, and Multi-Instance Learning framework into one unified model. We avoid the manually intensive work to label the text with distant supervision. To alleviate the problem of noisy labels, we model the event extraction task under the Multi-Instance Learning framework, wherein each entity is considered as a bag which consists of mentions of the entity. The LSTM module learns a distributed representation for the entity mention and is useful for encoding the dependencies between entity mentions in a sentence. To reduce the risk of overfitting, we design an auxiliary task of identifying event extent within the Multi-Task framework. We perform a comprehensive set of experiments on two real-world datasets and demonstrate that our approach significantly outperforms state-of-the-art baseline methods.

Chapter 6

Conclusion and Future Work

As the exponentially increasing amount of news articles being posted every day, it becomes more and more critical to develop algorithms to process the information from the text automatically. On the other hand, with the tremendous amount available data it also provides an excellent opportunity to utilize the invaluable event information in the news. In this dissertation, we investigated three data mining and information extraction problems in the domain of event detection, and encoding from newswire: (1) Comprehensively evaluate the quality of the extracted event from the large-scale event encoding system; (2) Detect interested event and identify the key sentences mentioning the story without sentence level labeling; (3) Encode event and extract event arguments with distant supervision.

6.1 Conclusion

To fully understand the challenges and pitfalls of the task of event extraction on the newswire, we start by an in-depth investigation of the current two well-known large-scale event extraction systems (ICEWS and GDELT) in political science domain. We investigate both systems from two different points of view: reliability analysis and validity analysis.

In the reliability analysis, we compute the Pearson Correlation between the protest event counts from ICEWS, GDELT, and GSR in daily, weekly and monthly level. Since GSR dataset is manually generated by human analysts, we consider it as the ground truth in our

experiments. The idea is that the higher correlation between the system's output and GSR indicates the better quality. Our results show that both ICEWS and GDELT have weak correlations with GSR in all three level. Furthermore, the correlation between ICEWS and GDELT is also at a very low level. These results suggest significant discrepancies between the outputs of automated systems and the hand-coded system.

In the validity analysis, we run a set of validations on both ICEWS and GDELT to evaluate the accuracy of these systems. First, we utilize the URL provided in the GDELT dataset and extract all the articles corresponding to the events. After keyword and temporal filtering, de-duplication operation, and a SVM based event classification, only 21% of the original GDELT records indicate a real protest event. Then, we apply the similar operations on ICEWS and around 80% of the keyword-filtered records are referring to real protest events. The common types of errors in both systems are Location Error, Polysemy Error, Planned Protest, Cancelled Protest, and Without time information. Duplication is also a significant pitfall in both systems. The duplication rates of each category of ICEWS and GDELT range between 10% and 30%.

We also do the further analysis on GDELT by manually check 3000 events from all 20 CAMEO categories(150 events for each group). The results show that the average accuracy over all categories is only around 16.2%. The most accurate category is Protest, which has the accuracy around 35.3%.

In short, while the efforts on event encoding are heroic, some predictable blind spots need to be addressed systematically in future event data efforts.

Inspired by the recent successful application of deep learning in Natural Language Processing, we proposed a MICNN model to detect the protest event and extract the key sentences from the news article simultaneously. To avoid the laborious work of labeling sentences, we model the problem under a multi-instance learning paradigm. We consider each news article as a bag, and each sentence is an instance. A hierarchical CNN is used to encode the local and global information for representation of the instance.

We design a compositional cost function which consists of four components: bag-level loss, instance ratio loss, instance-level loss, and manifold propagation loss. The bag-level loss is the major component which is a cross-entropy loss over entity bag labels. The instance ratio loss controls the number of positive instances being selected in each bag. The instance-level

loss is a variant of hinge loss which prefers the "maximum margin" solution. Moreover, the manifold propagation loss component is often used in unsupervised learning and make sure that the instances of similar representation would have same labels. The experiment results demonstrate the effectiveness of our model in protest event detection and key sentences extraction. The case study shows that our model could be able to capture the patterns of protest sentences, such as the mention of location and the existence of protest keywords. Through the highlighted example, we notice that the identified key sentences could be considered as the event summary and it could also help the human analysts quickly locate the essential information.

Finally, we extend the joint model into the domain of event encoding. Compared with sentence labeling, it is even much harder to label event arguments in word/phrase level manually. To avoid manually labeling the event arguments, we create the training dataset using distant supervision. One problem of applying distant supervision paradigm is that it assumes that if one entity takes a role of an event, then all the mentions of the entity in the article would take the same roles. It is easy to realize that this assumption is too strong.

To alleviate the problem caused by the strong assumption in distant supervision, we model the task as a multi-instance learning problem. Every entity defines a bag, and the bag consists of the mentions of the entity in the text. We follow the instance based paradigm and work on deriving the bag label from the instance' labels. In the context of event encoding, the labels of event arguments are heavily depended on each other. Follow this idea; we build the embedding of entity mention with a Bidirectional LSTM which encodes the dependencies between entity mention labels implicitly. Inspired by the idea of multi-task learning, we also design an auxiliary task of identifying event mention to reduce the risk of overfitting further. Results using two real world Cyber Attack and Protest datasets indicate that the proposed MIMTRNN model achieves better predictions than the strong baseline methods.

6.2 Future Work

As shown in this dissertation, the approach of combining representation learning and multi-instance learning demonstrates a promising result on the task of event detection and extraction. Through the representation learning module, we could avoid designing features

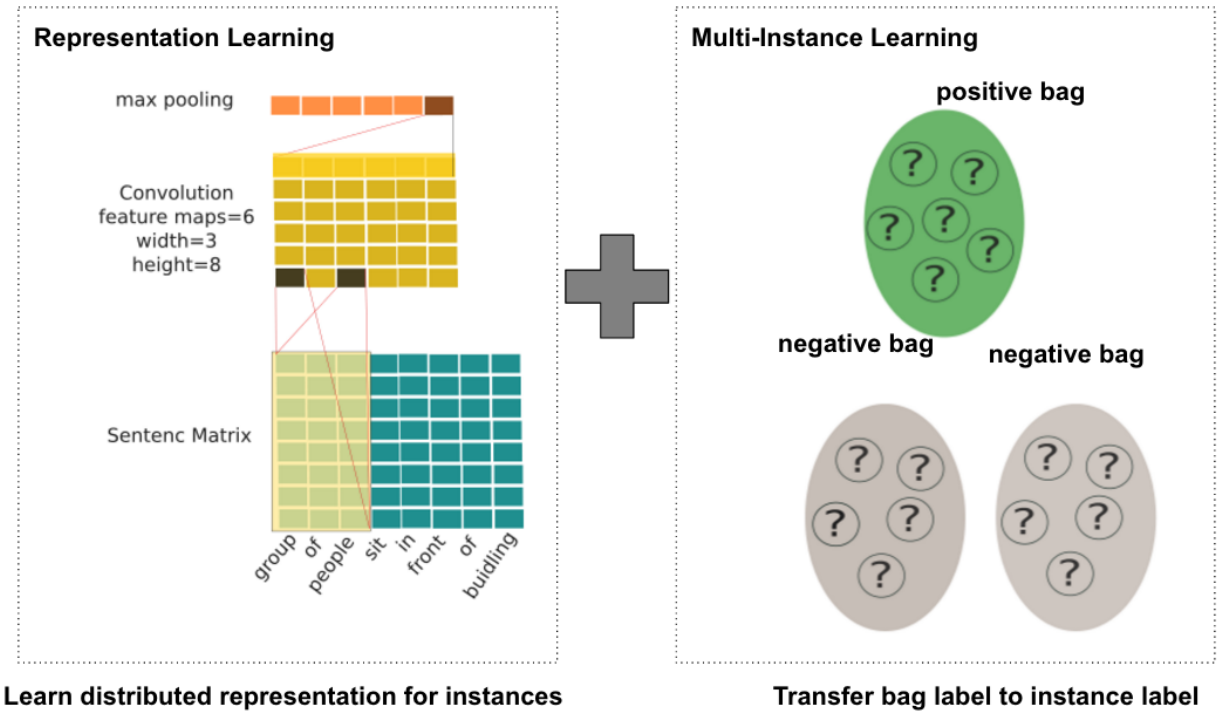

Figure 6.1: The joint framework of representation learning and multi-instance learning

manually and make the model easier to adapt to other domains. On the other hand, the multi-instance learning paradigm could allow the model being learned without finer level labels. The experimental results presented in this dissertation indicate several directions of future research on event extraction:

- Figure 6.1 shows the joint framework of representation learning and multi-instance learning. In this dissertation, we only tried two types of model (CNN and LSTM) for representation learning. The CNN model is good at detecting feature and capturing the spatial dependencies, while RNN does a good job at modeling the long distance dependencies. Inspired by the ideas from [21, 127], one promising direction is integrating the characteristics of event encoding into the process of representation learning. We will extend the representation learning module to exploit more types of neural network structure.

- In this dissertation, we followed the instance-based paradigm to solve the multi-instance learning problem. The process of deriving bag label only involves a small set of in-

stances and might ignore the useful information in the discarded instances. In future work, we will extend the multi-instance learning module to fully utilize all the instances in the bag. For instance, we could introduce the attention mechanism into MIL by modeling the bag probability as the weighted sum of instance probabilities. The weights are decided by a score function in which the parameters are learned by the model.

- Building the fully annotated training dataset is another challenge for the task of event extraction. On the other hand, it is relatively easy to annotate on document level that whether it mentions an event or not. It would be a good extension by integrating the semi-supervised learning and multi-task learning into a unified model. The semi-supervised learning module is designed to utilize both labeled and unlabeled data, and the multi-task module aims to improve the model by reducing the risk of overfitting. For instance, we could try to model the two tasks event detection (large dataset) and event encoding(small dataset) at the same time.

- In this dissertation, we only consider the events within the scope of one document. However, in the real world, one event is often reported by different news medias at the same time. Moreover, one news media might also publish a series of articles to follow up the updates of one event. Considering the event information from multiple sources might help the model make the robust prediction. As a future work, we plan to extend the scope of event extraction from one single article to multiple articles.

Bibliography

[1] examples of the use of gdelt in the policy community, including the world economic forum, and the washington post, among others, see the gdelt blog; http://blog.gdeltproject.org.

[2] The leading global thinkers of 2013: Innovators. foreign policy (2013); http://2013-global-thinkers.foreignpolicy.com/leetaru.

[3] Icews link in harvard dataverse; https://dataverse.harvard.edu/dataverse/icews, 2016.

[4] A. S. Abrahams, J. Jiao, W. Fan, G. A. Wang, and Z. Zhang. What's buzzing in the blizzard of buzz? automotive component isolation in social media postings. *Decision Support Systems*, 55(4):871–882, 2013.

[5] H. Adel, B. Roth, and H. Schütze. Comparing convolutional neural networks to traditional models for slot filling. *arXiv preprint arXiv:1603.05157*, 2016.

[6] A. Adi, D. Botzer, G. Nechushtai, and G. Sharon. Complex event processing for financial services. In *SCW*, pages 7–12. IEEE, 2006.

[7] J. Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence*, 201:81–105, 2013.

[8] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in neural information processing systems*, pages 577–584, 2003.

[9] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, pages 561–568. MIT Press, 2003.

[10] E. E. Azar. The conflict and peace data bank (copdab) project. *Journal of Conflict Resolution*, 24(1):143–152, 1980.

[11] A.-M. Barthe-Delanoë, S. Truptil, F. Bénaben, and H. Pingaud. Event-driven agility of interoperability during the run-time of collaborative processes. *Decision Support Systems*, 59:171–179, 2014.

[12] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. Theano: new features and speed improvements. NIPS, 2012.

[13] H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on twitter. *ICWSM*, 11(2011):438–441, 2011.

[14] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4):509–522, 2002.

[15] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[16] E. Boros, R. Besançon, O. Ferret, and B. Grau. Event role extraction using domain-relevant word representations. In *EMNLP*, pages 1852–1857, 2014.

[17] R. C. Bunescu and R. J. Mooney. Multiple instance learning for sparse positive bags. In *Proceedings of the 24th international conference on Machine learning*, pages 105–112. ACM, 2007.

[18] N. Chambers. Event schema induction with a probabilistic entity-driven model. In *Proc. EMNLP*, volume 13, pages 1797–1807, 2013.

[19] C.-Y. Chang, Z. Teng, and Y. Zhang. Expectation-regulated neural model for event mention extraction. page 400âĂŞ410.

[20] C. Chen and V. I. Ng. Joint modeling for chinese event extraction with rich linguistic features. In *In COLING*. Citeseer, 2012.

[21] Y. Chen, L. Xu, K. Liu, D. Zeng, J. Zhao, et al. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL (1)*, pages 167–176, 2015.

[22] H. Cheng, H. Fang, and M. Ostendorf. Open-domain name error detection using a multi-task rnn. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 737–746, 2015.

[23] R. Collobert and J. 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.

[24] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[25] M. Craven, A. McCallum, D. PiPasquo, T. Mitchell, and D. Freitag. Learning to extract symbolic knowledge from the world wide web. Technical report, DTIC Document, 1998.

[26] O. Data. The open event data alliance software page; https://openeventdata.github.io/.

[27] M. Denil, A. Demiraj, and N. de Freitas. Extraction of salient sentences from labelled documents. *arXiv preprint arXiv:1412.6815*, 2014.

[28] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1):31–71, 1997.

[29] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.*, 89(1-2):31–71, 1997.

[30] G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. Strassel, and R. M. Weischedel. The automatic content extraction (ace) program-tasks, data, and evaluation. In *LREC*, volume 2, page 1, 2004.

[31] L. Dong. *A comparison of multi-instance learning algorithms.* PhD thesis, The University of Waikato, 2006.

[32] G. Doran and S. Ray. A theoretical and empirical analysis of support vector machine methods for multiple-instance classification. *Machine Learning*, 97(1-2):79–102, 2014.

[33] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[34] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[35] X. Feng, L. Huang, D. Tang, B. Qin, H. Ji, and T. Liu. A language-independent neural network for event detection. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 66, 2016.

[36] J. R. Firth. A synopsis of linguistic theory, 1930-1955. 1957.

[37] C. C. for Democracy. Automatic document categorization for highly nuanced topics in massiveâĂŘscale document collections;http://www.clinecenter.illinois.edu/publications/speed-bin.pdf.

[38] C. C. for Democracy. Transforming textual information on events into event data within speedâĂİ (ccd, 2011);http://www.clinecenter.illinois.edu/publications/speed-transforming-textual-information.pdf.

[39] J. R. Foulds. *Learning instance weights in multi-instance learning*. PhD thesis, The University of Waikato, 2008.

[40] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola. Multi-instance kernels. In *ICML*, volume 2, pages 179–186, 2002.

[41] T. Gartner, P. A. Flach, A. Kowalczyk, and A. J. Smola. Multi-instance kernels. In *Proc. ICML*, pages 179–186, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[42] R. Ghaeini, X. Z. Fern, L. Huang, and P. Tadepalli. Event nugget detection with forward-backward recurrent neural networks. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 369, 2016.

[43] Google. World's largest event dataset now publicly available in bigquery (google, 2014); http://googlecloudplatform.blogspot.com/2014/05/worlds-largest-event-dataset-now- pulicly-available-in-google-bigquery.html.

[44] M. Hayes and P. Nardulli. Speeds societal stability protocol and the study of civil unrest: an overview and comparison with other event data projects (white paper). *Cline Center for Democracy. University of Illinois at Urbana-Champaign*, 2011.

[45] D. Heaven. World's largest events database could predict conflict. *New Scientist*, 218(2916):19–20, 2013.

[46] F. Herrera, S. Ventura, R. Bello, C. Cornelis, A. Zafra, D. Sánchez-Tarragó, and S. Vluymans. *Multiple Instance Learning: Foundations and Algorithms*. Springer, 2016.

[47] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proc. ACL*, pages 541–550. Association for Computational Linguistics, 2011.

[48] W.-T. Hsieh, T. Ku, C.-M. Wu, and S.-c. T. Chou. Social event radar: A bilingual context mining and sentiment analysis summarization system. In *Proc. ACL*, ACL '12, pages 163–168, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[49] R. Huang and E. Riloff. Modeling textual cohesion for event extraction. In *AAAI*, 2012.

[50] R. Huang and E. Riloff. Multi-faceted event recognition with bootstrapped dictionaries. In *HLT-NAACL*, pages 41–51, 2013.

[51] S. Jiang, H. Chen, J. F. Nunamaker, and D. Zimbra. Analyzing firm-specific social media and market: A stakeholder-based event analysis framework. *Decision Support Systems*, 67:30–39, 2014.

[52] X. Jiang, Q. Wang, P. Li, and B. Wang. Relation extraction with multi-instance multi-label convolutional neural networks.

[53] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[54] R. Johnson and T. Zhang. Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*, 2014.

[55] R. Johnson and T. Zhang. Deep pyramid convolutional neural networks for text categorization. 2017.

[56] M. I. Jordan. Serial order: A parallel distributed processing approach. *Advances in psychology*, 121:471–495, 1997.

[57] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.

[58] M. S. C. D. M. D. J. Kevin Reschke, Martin Jankowiak. Event extraction using distant supervision. In *Proceedings of LERC*, 2014.

[59] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[60] G. King and W. Lowe. An automated information extraction tool for international conflict data with performance as good as human coders: A rare events evaluation design. *International Organization*, 57(03):617–642, 2003.

[61] T. Korte. Creating a real-time global database of events, people, and places in the news; www.datainnovation.org/2013/12/creating-a-real- time-global-database-of-events-people-and-places-in-the-news/. 2013.

[62] D. Kotzias, M. Denil, N. de Freitas, and P. Smyth. From group to individual labels using deep features. In *Proc. SIGKDD*, KDD '15, pages 597–606, New York, NY, USA, 2015. ACM.

[63] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[64] K. Lee, Y. Artzi, Y. Choi, and L. Zettlemoyer. Event detection and factuality assessment with non-expert supervision. In L. MÃărquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton, editors, *Proc. EMNLP*, pages 1643–1648. The Association for Computational Linguistics, 2015.

[65] K. Leetaru. A new way to read the crisis in greece. foreign policy online, 31 july 2015; http://foreignpolicy.com/2015/07/31/greece-debt-syriza-tspiras-google/.

[66] K. Leetaru. Tracking the islamic stateâĂŤwith words. foreign policy online, 19 june 2015; http://foreignpolicy.com/2015/06/19/islamc-state-big-data-middle-east/.

[67] K. Leetaru and P. A. Schrodt. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA Annual Convention*, volume 2. Citeseer, 2013.

[68] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[69] P. Li, G. Zhou, Q. Zhu, and L. Hou. Employing compositional semantics and discourse consistency in chinese event extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1006–1016. Association for Computational Linguistics, 2012.

[70] Q. Li, H. Ji, and L. Huang. Joint event extraction via structured prediction with global features. In *ACL (1)*, pages 73–82, 2013.

[71] R. Lin, S. Liu, M. Yang, M. Li, M. Zhou, and S. Li. Hierarchical recurrent neural network for document modeling. In *EMNLP*, pages 899–907, 2015.

[72] Y. Lin, S. Shen, Z. Liu, H. Luan, and M. Sun. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*, volume 1, pages 2124–2133, 2016.

[73] G. Liu, J. Wu, and Z.-H. Zhou. Key instance detection in multi-instance learning. In S. C. H. Hoi and W. L. Buntine, editors, *ACML*, volume 25 of *JMLR Proceedings*, pages 253–268. JMLR.org, 2012.

[74] W. Lu and D. Roth. Automatic event extraction with structured preference modeling. In *Proc. ACL*, ACL '12, pages 835–844, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[75] W. Lu and D. Roth. Joint mention extraction and classification with mention hypergraphs. In *EMNLP*, pages 857–867, 2015.

[76] M. Maire, T. Narihira, and S. X. Yu. Affinity cnn: Learning pixel-centric pairwise relations for figure/ground embedding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[77] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *Advances in neural information processing systems*, pages 570–576, 1998.

[78] D. McClosky, M. Surdeanu, and C. D. Manning. Event extraction as dependency parsing. In *Proc. ACL*, HLT '11, pages 1626–1635, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[79] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu, et al. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(3):530–539, 2015.

[80] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[81] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Hlt-naacl*, volume 13, pages 746–751, 2013.

[82] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proc. ACL*, pages 1003–1011. Association for Computational Linguistics, 2009.

[83] S. Muthiah, B. Huang, J. Arredondo, D. Mares, L. Getoor, G. Katz, and N. Ramakrishnan. Planned protest modeling in news and social media. In *AAAI*, pages 3920–3927, 2015.

[84] K.-H. Nguyen, X. Tannier, O. Ferret, and R. Besançon. Generative event schema induction with entity disambiguation. In *Proc. ACL*, 2015.

[85] T. H. Nguyen, K. Cho, and R. Grishman. Joint event extraction via recurrent neural networks. In *Proceedings of NAACL-HLT*, pages 300–309, 2016.

[86] T. H. Nguyen and R. Grishman. Event detection and domain adaptation with convolutional neural networks. *Volume 2: Short Papers*, page 365, 2015.

[87] J. Nothman, M. Honnibal, B. Hachey, and J. R. Curran. Event linking: Grounding event reference in a news archive. In *Proc. ACL*, ACL '12, pages 228–232, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[88] W. Nuij, V. Milea, F. Hogenboom, F. Frasincar, and U. Kaymak. An automated framework for incorporating news into stock trading strategies. *IEEE transactions on knowledge and data engineering*, 26(4):823–835, 2014.

[89] S. P. O'brien. Crisis early warning and decision support: Contemporary approaches and thoughts on future research. *International Studies Review*, 12(1):87–104, 2010.

[90] G. Palmer, V. dâĂŹOrazio, M. Kenwick, and M. Lane. The mid4 dataset, 2002–2010: Procedures, coding rules and description. *Conflict Management and Peace Science*, 32(2):222–242, 2015.

[91] B. A. Pearlmutter. Learning state space trajectories in recurrent neural networks. *Neural Computation*, 1(2):263–269, 1989.

[92] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[93] A.-M. Popescu, M. Pennacchiotti, and D. 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.

[94] N. Ramakrishnan, P. Butler, S. Muthiah, N. Self, R. Khandpur, P. Saraf, W. Wang, J. Cadena, A. Vullikanti, G. Korkmaz, et al. 'beating the news' with embers: forecasting civil unrest using open source indicators. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1799–1808. ACM, 2014.

[95] N. Ramakrishnan, P. Butler, S. Muthiah, N. Self, R. Khandpur, P. Saraf, W. Wang, J. Cadena, A. Vullikanti, G. Korkmaz, et al. 'beating the news' with embers: Forecasting civil unrest using open source indicators. In *Proc. SIGKDD*, pages 1799–1808. ACM, 2014.

[96] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[97] M. Rei. Semi-supervised multitask learning for sequence labeling. *Proceddings of ACL*, 2017.

[98] S. Riedel, L. Yao, and A. McCallum. Modeling relations and their mentions without labeled text. *Machine learning and knowledge discovery in databases*, pages 148–163, 2010.

[99] A. Ritter, O. Etzioni, S. Clark, et al. Open domain event extraction from twitter. In *Proc. SIGKDD*, pages 1104–1112. ACM, 2012.

[100] A. Ritter, E. Wright, W. Casey, and T. Mitchell. Weakly supervised extraction of computer security events from twitter. In *Proceedings of the 24th International Conference on World Wide Web*, pages 896–905. ACM, 2015.

[101] G. Rizzo and R. Troncy. Nerd: a framework for unifying named entity recognition and disambiguation extraction tools. In *Proc. ACL*, pages 73–76. Association for Computational Linguistics, 2012.

[102] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[103] P. A. Schrodt, J. Beieler, and M. Idris. ThreeâĂŹsa charm?: Open event data coding with el: Diablo, petrarch, and the open event data alliance. In *ISA Annual Convention*, 2014.

[104] P. A. Schrodt and D. J. Gerner. Analyzing international event data: A handbook of computer-based techniques. 2012.

[105] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil. Learning semantic representations using convolutional neural networks for web search. In *Proc. WWW*, pages 373–374. International World Wide Web Conferences Steering Committee, 2014.

[106] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE, 2003.

[107] V. Subrahmanian. *Handbook of computational approaches to counterterrorism*. Springer Science & Business Media, 2012.

[108] V. Subrahmanian, M. Albanese, M. V. Martinez, D. Nau, D. Reforgiato, G. I. Simari, A. Sliva, J. Wilkenfeld, and O. Udrea. Cara: A cultural-reasoning architecture. *IEEE Intelligent Systems*, 22(2):12–16, 2007.

[109] M. Surdeanu, S. Gupta, J. Bauer, D. McClosky, A. X. Chang, V. I. Spitkovsky, and C. D. Manning. Stanford's distantly-supervised slot-filling system. In *TAC*, 2011.

[110] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning. Multi-instance multi-label learning for relation extraction. In *Proc. EMNLP*, pages 455–465. Association for Computational Linguistics, 2012.

[111] T. Tieleman and G. Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. Technical report, Technical report, 2012. 31.

[112] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, 2002.

[113] J. C. Van Gemert, C. J. Veenman, A. W. Smeulders, and J.-M. Geusebroek. Visual word ambiguity. *IEEE transactions on pattern analysis and machine intelligence*, 32(7):1271–1283, 2010.

[114] N. T. Vu, H. Adel, P. Gupta, and H. Schütze. Combining recurrent and convolutional neural networks for relation classification. *arXiv preprint arXiv:1605.07333*, 2016.

[115] D. Z. Wang, Y. Chen, S. Goldberg, C. Grant, and K. Li. Automatic knowledge base construction using probabilistic extraction, deductive reasoning, and human feedback. In *Proc. ACL*, pages 106–110. Association for Computational Linguistics, 2012.

[116] J. Wang and J.-D. Zucker. Solving multiple-instance problem: A lazy learning approach. 2000.

[117] N. Weidmann, E. Frank, and B. Pfahringer. A two-level learning method for generalized multi-instance problems. In *Proc. EMNLP*, pages 468–479, 2003.

[118] N. B. Weidmann and M. D. Ward. Predicting conflict in space and time. *Journal of Conflict Resolution*, 54(6):883–901, 2010.

[119] J. Weston, A. Bordes, S. Chopra, and T. Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

[120] D. Wurzer, V. Lavrenko, and M. Osborne. Twitter-scale new event detection via k-term hashing. In L. MÃărquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton, editors, *Proc. EMNLP*, pages 2584–2589. The Association for Computational Linguistics, 2015.

[121] X. Xu. *Statistical learning in multiple instance problems.* PhD thesis, University of Waikato, 2003.

[122] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, pages 1480–1489, 2016.

[123] K. Yao, G. Zweig, M.-Y. Hwang, Y. Shi, and D. Yu. Recurrent neural networks for language understanding. In *Interspeech*, pages 2524–2528, 2013.

[124] J. Yu and J. Jiang. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. pages 236âĂŞ–246, 2016.

[125] M. Yu, M. R. Gormley, and M. Dredze. Combining word embeddings and feature embeddings for fine-grained relation extraction. In *Proc. NAACL*, 2015.

[126] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[127] D. Zeng, K. Liu, Y. Chen, and J. Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*, pages 1753–1762, 2015.

[128] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pages 13–13. IEEE, 2006.

[129] Q. Zhang and S. A. Goldman. Em-dd: An improved multiple-instance learning technique. In *Advances in neural information processing systems*, pages 1073–1080, 2002.

[130] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

[131] A. Zhila and A. F. Gelbukh. Open information extraction for spanish language based on syntactic constraints. In *ACL (Student Research Workshop)*, pages 78–85, 2014.