An Online Structured Political Event Dataset based on CAMEO Ontology

Sayeed Salam, Patrick Brandt, Vito D'Orazio, Jennifer Holmes, Javiar Osorio, Latifur Khan
University of Texas at Dallas, The University of Arizona
(sxs149331, pbrandt, dorazio, jholmes, lkhan)@utdallas.edu, josorio1@email.arizona.edu

Abstract

Political activities and interactions between different global entities are becoming growing field for data-intensive computing with a wide scope of research opportunities for both social science and computer science researchers. This research needs to be carried out at a local (limited to a particular region) and global scale, often divided in temporal manner. It is also useful to have the most recently updated dataset for relevant analysis. For these purposes, we need timestamped, geolocaated structured information about political interactions. Keeping this in mind, we develop a datatset that complies with Conflict and Mediation Event Observation (CAMEO) ontology inspired by the "who-did-what-to-whom" format. We use a distributed framework for data collection and processing that works in real-time with Apache Kafka and SPARK in order to process a global collection of news data in different languages (i.e., Spanish, Arabic) and generate those structured event data in real-time. We also provide an API for easy access to the data. In this paper, we describe how the data is represented, collected, and processed, how we generate the most up-to-date dataset with dynamic ontology extension, and how to access the data and possible analytical problems that can be addressed by building a model on the dataset.

Introduction

Political interaction around the world is rooted in a convoluted network of information. Due to globalization, two entities or nations can be politically more related and be easily analyzed with the appropriate interactions captured in some structured form. This type of data generation is considered one of the key areas of information extraction from political interactions. This process involves converting unstructured text data to computer-friendly structured events. Most of the time the unstructured data consists of online media content and text articles published by different news agencies across the world. The encoding process (from unstructured text to structured data) is done either with the help of a machine learning-based classifier or fixed ontologies (Khan and McLeod 2000)(Schrodt 2012a). These are used to identify critical information from a sentence - for example, who is acting, who is being acted upon, and what type of action is taking in place. Between these two approaches, the latter provides better granularity with reasonable accuracy and simplicity. One example of fixed ontologies is CAMEO-Conflict and Mediation Event Observations (Gerner et al. 2002) and is defined by dictionaries as verb patterns and political entities. There are automated event coders like PETRARCH (Norris, Schrodt, and Beieler 2017) and UD-PETRARCH which are used to generate events using pattern matching and entity lookup from these dictionaries. The information organized by the automated coders is known as the *who-did-what-to-whom* format.

Both versions of PETRARCH event coders rely on some form of metadata at the sentence level (i.e. Parts-Of-Speech Tags, Universal Dependency parses, etc.) and require compute-intensive processing. To accomadate for this, we use distributed solutions for dividing the workload in Apache Spark based clusters.

News articles are continuously being published around the world. If we want to get a real-time comprehension of multiple simultaneous political interactions ocurring around the globe, we need to build a real time data processing framework where we can collect data in real-time, process it, and present it to the user for visualization (D'Orazio, Deng, and Shoemate 2018) and analysis. Unavailability of such a system hinders researchers in the progression of their research and undermine the efficiency of ontologies built to capture this information.

Easy access to the dataset is important for obtaining the required data for processing. People from all levels of technical background should be able to easily access data with little to no training on the system.

Focusing on these key points, we build an infrastructure to manage and generate metadata with the help of a real-time data processing framework that collects news articles from the web, processes them with Stanford CoreNLP/Universal Dependency parsers, and generates events using PETRARCH event coder. We also build an API for easy access to the generated events in real-time.

In this paper, we will discuss the following aspects

- Dataset and metadata description
- Real-time data generation process (Apache SPARK and Kafka)
- Access to the generated data

- Some example scenarios where analytical models can be built using the data we have

Background

To help the reader better understand the tools and methods used in this paper, we provide key details of fundamental concepts.

Political Events are structured pieces of information consisting of an action, source (acting entity), target (entity being acted upon) and other related information. For example, consider the following extract from a news article -

```
PM Theresa May has struck a last-minute
deal with the EU in a bid to move Brexit
talks on to the next phase.
```

As a structured event, it looks like the following

```
Source - GBRGOV
Target - IGOEUREEC
Action - 057 (Sign formal agreement)
```

PM Theresa May and EU are coded in standard format for event generation. The structure used here is mostly known as the *who-did-what-to-whom* format of event coding. The coding mechanism is depicted in the Figure 1

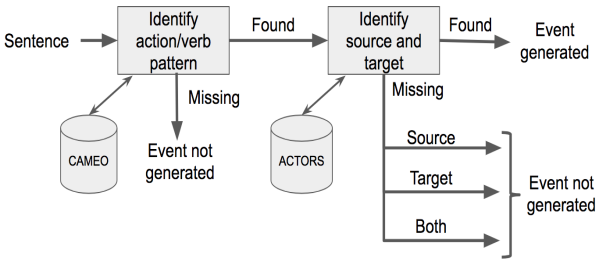

Figure 1: Basic Mechanism of Automated Coding using PETRARCH

Given a sentence, the encoder searches for a matching pattern in the CAMEO verb patterns dictionary. A pattern consists of a verb and surrounding keywords. Together they signify a particular course of action and are represented by event code. For example, SET in the pattern "SET OUT VIEWS" indicates a MAKE PUBLIC STATEMENT type of event (event code 010). Upon finding a match in pattern, actor dictionaries search for matching entities representing the source and target. After finding the necessary information, an event is coded by PETRARCH. If there is missing information, PETRARCH will ignore the event. These sequence of events are called Source-Action-Target or SAT format.

Conflict and Mediation Event Observations (CAMEO) is an ontology developed to capture political events and focuses primarily on the following 4 categories

- Verbal Cooperation - when two entities are agreeing or co-operating verbally on a matter.
- Material Cooperation - when an entity (source) is actively helping other entity (target)
- Verbal Conflict - when two entities are in disagreement on a matter.
- Material Conflict - when an entity (source) are in conflict physically with another entity (target), i.e., protest, war, etc

It works using a knowledge-base of pattern and actor dictionaries. Pattern dictionaries are helpful for identifying political interactions in a given sentence. Actor dictionaries are used to search for political actors around the matched pattern in a sentence.

mai

Structure of Event Data

We have collected more than 1.3 million political events by using automated event coders (i.e. Petrarch, UD-Petrarch). Each of the events are timestamped and geolocated based on information in the corresponding news article. We identified the source, target, and action in compliance with CAMEO's who-did-what-to-whom format. Here are the key attributes for the dataset-

- "Code" represents type of event at the finer level
- "Root Code" is derived from "Code" at a less finer type specification of events
- "Quad Class" is the most coarse representation of event type, cosidered as primary taxonomy of CAMEO.
- "Source" represents the entity acting.
- "Target" represents the entity being acted upon.
- "Date8" is the date of the event in YYYYMMDD format.
- "Geoname" identifies the location of the event.
- "Latitude" and "Longitude" are the geo-coordinates for the place where the event took place.
- "Goldstein Score" provides WEIS Compatible score (Goldstein 1992) for event severity/polarity
- "Source Text" identifies the source of the news article.
- "URL" provides WWW location for the article
- "Id" uniquely identifies the document and the sentence generated the event

Other attributes found in the dataset can be derived from the attributes using rules defined in ontology. For example, here is an event specified in JSON format.


```
{
"_id" :
    ObjectId("59f7ffe8583dca26c72947f9"),
"code" : "020",
"src_actor" : "USA",
"month" : "10",
"tgt_agent" : "",
"country_code" : "USA",
"year" : "2019",
"id" : "59f7fea8de7923402d8a5df9_1",
"source" : "USAGOV",
"date8" : "20190815",
"src_agent" : "GOV",
```

```
"tgt_actor" : "CHN",
"latitude" : 95.7129,
"src_other_agent" : "",
"quad_class" : 1,
"source_text" : "bbc",
"root_code" : "02",
"tgt_other_agent" : "",
"day" : "15",
"target" : "CHNGOV",
"goldstein" : 0,
"geoname" : "United States",
"longitude" : 37.0902,
"url" : "https://www.bbc.com/news/
world-asia-china-49353727",
"date8_val" :
    ISODate("2019-08-15T00:00:00Z")
}
```


The event correspons to the following sentence.

Sentence 1: *US President Donald Trump has urged Xi Jinping to meet protesters in Hong Kong who have been demanding democratic reforms.* [1]

In later sections, we use this sentence for illustration purposes. For a detailed description about each field of the above event, please refer to the CAMEO codebook (Schrodt 2012a). Specifically to our purpose, the three key fields of information are the source (USAGOV), the target (CHNGOV) and the type of event (code=02 means make an appeal or request of generic type).

Event Coding Framework

To sustain the dataset, we designed an infrastructure to download articles from the web, process them to generate metadata, and run event coding and geolocation algorithms followed by the distribution of the data using a web based API. The following subsections will describe each stage in detail from the framework.

Figure-2 depicts the components of the infrastructure and how the pipeline works. The modules with white background are part of a real-time system. Other modules with a gray background are experimental and geared towards Spanish article processing. From the web we collect Spanish news articles using Web Crawler (Crawler) and English news articles using Web-Scraper(Scraper). Then we filter out the Spanish articles and keep only the politically relevant articles using ML based filtering classifier (Filter). Filtered documents are passed through the text Processing module where Universal Dependency parse trees are generated at the sentence level (Text Processing). The processed output is provided to the Political Event Coding module where UD-PETRARCH (Political Event Coding) and geo-location software work together to generate geo-located events (events along with the place where it happened). The generated events are then distributed using the API to the Researchers and Visualization tools (API and Visualization). Among the Spanish article related modules, we create a classifier model (Model Building) using annotated data. The annotation module supports multilingual validation where we generate a parallel corpus language-wise (in English and Spanish) and observe the generated events from both sets of documents. In the following paragraphs, we discuss about the components of online political event coding in detail. The infrastructure essentially has five components.

- Data Collection and Filtration
- Data Processing and Meta-data Generation
- Event Generation and Data Distribution
- Annotated dataset generation and Quality Assessment of Generated Event Data
- Ontology translation to foreign language (i.e., Spanish) from English

The first three components are part of the real-time political event coding process. The rest of them are related to offline modules that support the real-time coding process.

Article collection: Crawler and Scrpaer

We collect English articles using scraping 250+ well defined RSS Feeds (whi 2016) that list political news articles from different news agencies worldwide. We run the scraper daily with a 20 minute interval.

For Spanish news articles, we run a crawler using a list of news agencies from Spain, South America, and other Spanish-speaking countries.

These articles that are downloaded daily become the input for the framework. We then run the event coding process to generate geolocated events in near real-time and it can be served through the web API in real-time depending on the query parameter. A detailed list of Spanish news websites can be found here (OEDA 2018).

Article Annotation

This step is required for Spanish news articles as we download those articles without topic restrictions and the websites often did not present a way to collect data section wise. Sometimes, a news article may also have URLs to other news articles that the crawler can follow and download. After the data collection, we will get a lot of articles that are not relevant and of those that are relevant, not all of them are political. For this we annotate a subset of articles with randomly chosen 450+ documents and annotate whether they are relevant, and if so, whether they are focused on politics. Once an article is found to be politically relevant, we classify it into quad-class categories of the CAMEO ontology and thus create a simplified Gold Standard Records (GSR). This set of articles will be used for validating the accuracy/coverage of the event coder.

Article Filtration - Relevant/Irrelevant followed by political/non political

We have designed a machine learning algorithm to filter out articles that are not relevant to politics (either irrelevant articles or articles from other sections like business, sports, etc). As we have found from the sample set in the annotation step, only 35-45% of the downloaded articles are relevant. Among them, around 50% of the articles talk about politics. We train

[1] https://www.bbc.com/news/world-asia-china-49353727

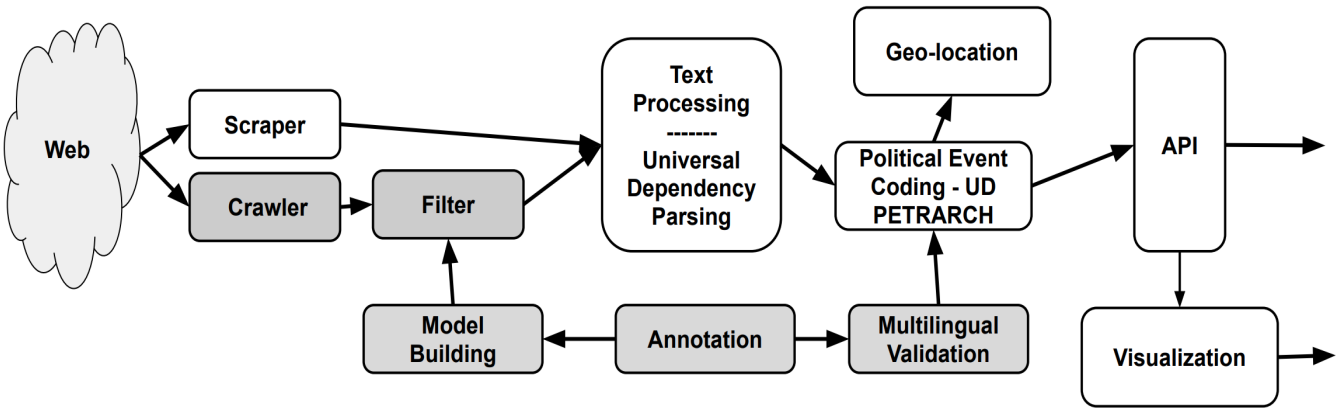

Figure 2: Framework Diagram

the model using the annotated documents and test it on a set of 200 annotated documents. We have used TF-IDF on Bi-grams for feature selection technique and RandomForest, NaiveBayes, and SVM as classifiers.

Universal Dependency Parse Generation

After filtering out the irrelevant articles, we process the remaining documents to generate metadata from raw text. The metadata includes Parts-of-Speech(POS) tags, named-entity tags along with the dependency relationships. We use ufal-udpipe python package to generate the tags and relationship between words at a sentence level. For an example, consider the Sentence 1.

The corressponding dependency parse along with POS tags are given below -

```
1 US US PROPN NNP Number=Sing 2
compound _ _
2 President President PROPN NNP
Number=Sing 6 nsubj _ _
3 Donald Donald PROPN NNP Number=Sing 2
flat _ _
4 Trump Trump PROPN NNP Number=Sing 2
flat _ _
5 has have AUX VBZ Mood=Ind|
Number=Sing|Person=3|Tense=Pres|
VerbForm=Fin 6 aux _ _
6 urged urge VERB VBN
Tense=Past|VerbForm=Part 0 root _ _
7 Xi Xi PROPN NNP Number=Sing 6 obj _ _
8 Jinping Jinping PROPN NNP Number=Sing
7 flat _ _
. . .
. . .
```

The format used here is called CoNLL-U format and details related to attributes per line can be found here (Muniz, Chalub, and Rademaker 2017).

The corresponding dependency tree is shown in Figure 3 where each node in the tree can be seen.

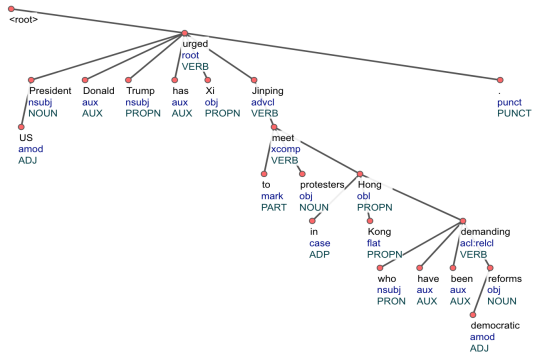

Figure 3: Universal dependency tree for English

For each sentence in a document, we generate a dependency parse tree. As shown in the analysis section, the parsing job requires lot of time to complete when run in standalone mode. We have adapted a Spark-based distributed system for this task as it speeds up the processing almost linearly with an increase in the number of processing cores. Each Spark worker node generates the dependency parse for sentences from a batch of news articles and stores them in a NoSQL database like MongoDB (provides better scalalbility).

Event Coding

After generating metadata, we process them with UD-PETRARCH event coder (Lu 2019) to generate time-stamped political events. It generates CAMEO compatible events. For geo-locating events, we use Cliff-Clavin (D'Ignazio et al. 2014) from MediaMeter. It gives us locations associated with the events found in the sentences. Here is a brief description on how event coding process works for a particular sentence. Given the sentence with dependency relations, the coder first identifies the noun and verb phrases of the sentence. For the example sentence, we have them listed in Table 1.

The coder then identifies the root verb, which is "urge" in this case. Then using the dependency relationship between

Noun Phrase	Verb Phrase
President Donald Trump, Xi Jinping, protesters, Kong, democrate, reforms	urged, meet demanding

Table 1: Noun and Verb Phrases for Sentence 1

noun phrases and the root verb, the coder creates triplets in the form of (source, action, target). An example triplet that eventually qualifies for an event is as follows:

```
'matched_txt': '- * + [020]
#line:12822',
'source_text': 'US PRESIDENT
DONALD TRUMP',
'target_text': 'XI JINPING',
'verb_text': 'URGE',
'verbcode': '020'
```

Here the "matched_text" is the verb pattern in the sentence that has been found in the CAMEO verb dictionary. The source and target are matched against the actor dictionaries and once all the information are found, the triplet becomes an event.

All the processed text articles and generated events are stored in the MongoDB. All these framework components use Apache Kafka to synchronize the inputs and outputs.

Multilingual Event Coding (Experimental)

We are using UD-PETRARCH event coder that works on universal dependency rather than only the Parts-Of-Speech tags used by original version of PETRARCH. It is helpful for multilingual event coding because of the uniformity of universal dependency parses across language in comparison to the inconsistency of Parts-of-Speech tags across languages. Using a universal dependency based event coder, we can easily incorporate coding in other languages with no effort needed on updating the event coder itself. We will still need the dictionaries to be translated first. Using a universal dependency parser, however, gives flexibility for the non-CS background researchers to solely focus on the analysis and ontology extension parts. An example of universal dependency is presented in the following sentence. Now, let us consider the dependency graph of Sentence 1, after translating it to Spanish.

Sentence 1 (Spanish):*El presidente de los Estados Unidos, Donald Trump, instó a Xi Jinping a reunirse con manifestantes en Hong Kong que han estado exigiendo una reforma democrática*

Figure 4 shows how the dependency relations are structured between words in the sentence.

Once again the root verb here is "instó", meaning "urge" in English. This time UD-PETRARCH genrates the following triplet

```
'matched_txt': u'- * +        [020]
#line:18266',
'source_text': 'EL PRESIDENTE DE LOS
ESTADOS UNIDOS',
'target_text': u'XI JINPING',
'verb_text': u'INSTAR',
```

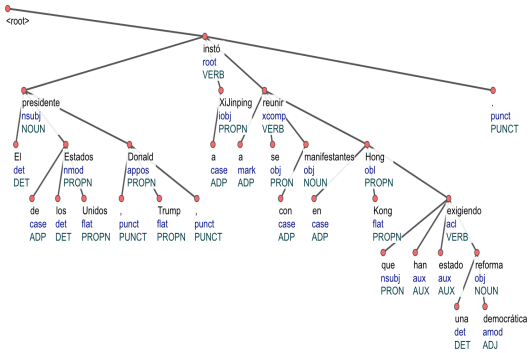

Figure 4: Universal dependency tree for Spanish

```
'verbcode': u'020'
```

and we found the same event where the source is USAGOV, the target is CHNGOV, and the event type code is 020.

Cross-lingual validation for generated events.

To identify whether similar events reported in different languages can be captured by the event coder, we run the follwoing validation procedure. First, we select a set of documents in Spanish that has been annotated to be politically relevant. Then we translate them to English using Google Translate API. After this step we have a parallel corpus of english and spanish documents. Then we run respective language parsers to generate universal Dependency reationships and feed them to the UD-PETRRACH event coder. We observe the genrated triplets and do a semantic comparison between reported source and target text with the help of BabelNet. We also follow whether they are reporting the same event type code. We obsereved strong similarity between the generated events.

Ontology Translation From English

To capture events within articles published in Spanish or in another foreign language, we have to translate the existing CAMEO ontology to the corresponding language. There are two parts that needs to be translated at this step. Those are verb-patterns and the list of political entities (political leaders, organizations, etc.).

Translating verb patterns: Verb patterns are used to capture the interaction between two entities. To translate verb-patterns, we adopted a semi automated way and developed an online application to facilitate the collaboration between human annotators. We first presented a basic translation of verbs using the Wordnet (Fellbaum 1998) synsets in Spanish. The human annotators were asked to access the quality of translation w.r.t the CAMEO code category and after all the feedbacks, we did a majority analysis to include a translation in the new dictionary (Osorio et al. 2019).

Translating CAMEO entities: With this approach we will translate CAMEO dictionaries containing political entities and organizations. The algorithm translates dictionaries written in one language to another specified language using a

database of translations from BableNet source. We observed that the Bablenet translation database is more accurate for translating agents, actors, countries, and organization names than other translation sources such as Google and JRC databases (Steinberger et al. 2013). The Bablenet translator helped overcome problems that arise from other translators such as ambiguous and non-existing translations because it provides multiple translations for each entry since it gathers different possible translations from different sources. The process takes a dictionary file written in a specific language as an input and produces a translated version. It goes through each of the entities and their synonyms and translate them using BabelNet database (Navigli and Ponzetto 2012) to the specified target language. We get several translations and each of them are associated with scores. Translations with the highest scores are considered first. If no translation on BabelNet is present, Google Translate is used to do the translation task.

In the case of translating a synonym, the returned translated synonym set is clustered with other translated synonym sets belonging to the same main Entity. The algorithm leverages Levenshtein distances technique (Miller, Vandome, and McBrewster 2009) to calculate the similarity between each synonym pair in the synonym set. Then the algorithm creates a two dimensional array that stores the Levenshtein distance between each pair of synonyms in the set. After that, the algorithm builds a tree structure diagram of possible clustering based on hierarchical clustering techniques (Müllner 2011). After that, the elbow method (Wikipedia contributors 2019) is applied to determine the optimal number of clusters based on the hierarchical tree.

Online Ontology Extension

As we started using CAMEO ontology we found it to be static, moderately updated and often missing current information on political entities and interactions. To overcome this we use RePAIR framework to identify potential new political entities (Solaimani et al. 2017). We also apply the text mining approaches for identifying new political interactions that can be added to the current dictionary (Skorupa Parolin et al. 2019).

API Access to Event Data

We provide API to serve generated event data. This API maintains API-key based access restrictions and serves the data in JSON format. Interested users can query using a defined query language (similar to MongoDB JSON query language). Users can select the portion of the data they are interested in by subsetting the data. They can also focus on particular fields on each record, aggregate/group the query results, and query for the metadata about the datasets. Additionally, there are user defined libraries in R to make the data access easier (Kim et al. 2019) which is built on top of the web based API. Details of the access policy can be found here (Salam et al. 2020)

System Configuration

The framework components are running in different nodes of JetStream cloud under XSEDE, NSF initiative to facilitate academic research (Stewart et al. 2015). There we have in total of 7 nodes. Three of them forms a Spark Cluster with 30 cores and 90 GB of ram. We have a 6-core machine to collect data from website using the scraper program. We run a single-instance of Cliff-Clavin server on a 10 core machine along with the PETRARCH event coder. We have a separate 6 core machine for running other utility servers like Apache Kafka, Zookeeper, etc. We also have a dedicated machine for MongoDB Storage Engine. Each node in there is Dell's M630 machines.

Dataset Description

Using the web-based API, we are currently serving the real-time dataset and other prominent event datasets. Those are listed in the Table-2

Event Dataset	Timespan	Count
ICEWS	1995 - Sep 2016	15,220,347
Cline Phoenix NYT	1945-2005	1,092,211
Cline Phoenix FBIS	1995-2004	8,179,55
Cline Phoenix SWB	1979-2015	2,906,715
TERRIER	1979-2016	68,233,154
Phoenix Real-Time	Oct 2017 - ongoing	1,315,369+

Table 2: List of available datasets

The ICEWS dataset (Boschee et al. 2018) comprises of events related to politics and has built a system off of the dataset to provide early warnings about political conflict. The Cline center dataset (Althaus and Shalmon 2017) covers Phoenix event data for the period of 1945 to 2015. They used PETRARCH-2 as the event coder and have done event coding on documents from New York Times (NYT), BBC Monitoring's Summary of World Broadcasts (SWB), and CIA's Foreign Broadcast Information Service (FBIS). TERRIER (Halterman et al. 2017b) is a datatset built on the previous versions of the offline data pipeline that use PETRARCH event coder. Phoenix Real-Time is the dataset being accumulated by our current infrastructure using up-to-date English news articles.

Real-time Dataset Summary

In this section we provide some basic analysis of the dataset generated by the framework to give the user some insight about using the data. Results presented here are calculated on the entire dataset collected from October 2017 to August 2018. Up until now, we have collected close to 5.5 million English news articles from 250+ news sources(whi 2016). Figure-5 shows Top-20 news sources representing 70% of the articles collected. Over 1.3 million political events have been generated from these articles. Figure-6 reflects the distribution of events based on root code(i.e 01, 02,...,20). Each indicates a particular category of events, more granular than quad class based event distribution. Here, we found MAKE PUBLIC STATEMENT (root code is 01) has the highest

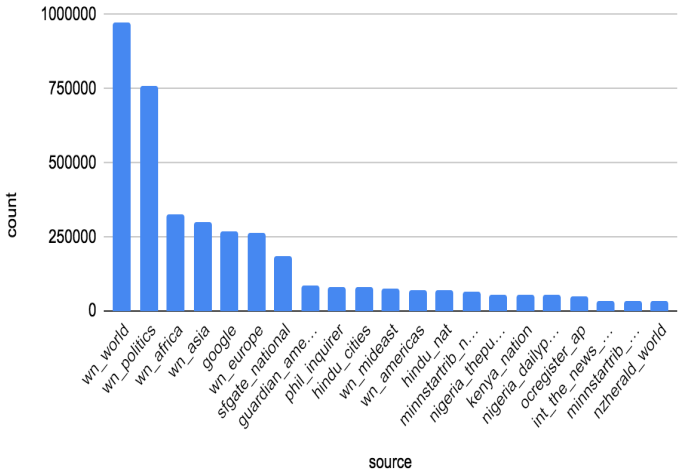

Figure 5: Top-20 news sources w.r.t English news articles

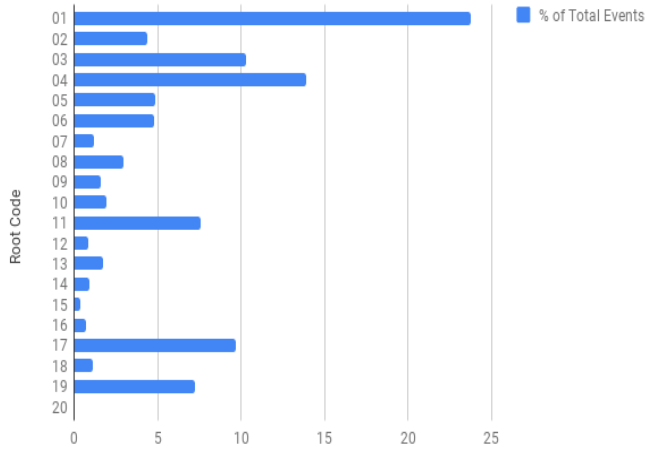

Figure 6: Distribution of events based on root code

percentage followed by CONSULT (root code=04). This is because a large number of the news articles have a figure making a public statement in the beginning of the text. PETRARCH2 focuses on the first paragraph (up-to 6 sentences) for event coding. That is why there is higher percentage of events in that category. We consider the root-code 01 for further granularity and gather the percentage of events belonging to each of the 10 event code (i.e 010, 011,...,019.) Again we discovered a highly imbalanced distribution where event code 010 has largest percentage of $\approx 87\%$. It indicates that the most prevalent event is a public statement expressed verbally or in action not otherwise specified. In the following example, we try to derive the political relation between two countries Saudi Arabia (SAU) and QATAR (QAT). We plot the quad class of the events where source is SAU and target is QAT and Figure-8 reflects that. There are 224 events and almost 95% of them indicate a non-friendly relationship between these two entities. This also correlates with the real

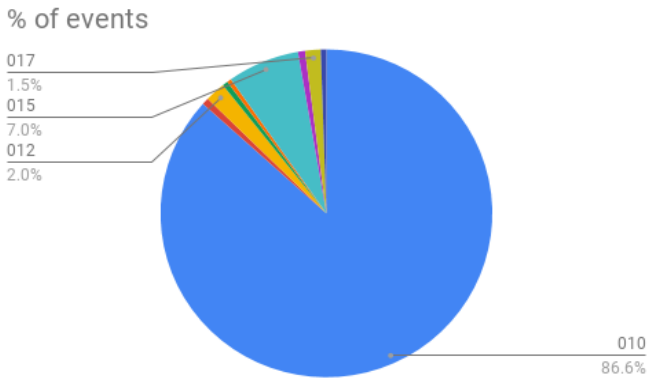

Figure 7: Distribution of events with root code 01

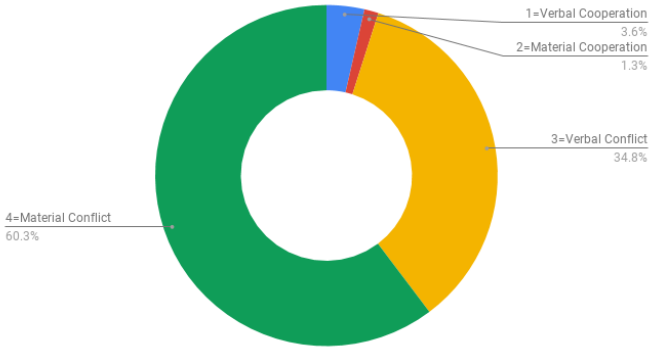

Figure 8: Distribution of quad classes where source=SAU and target=QAT

world scenario where we observe political conflict between these two countries including terrorism and airline embargos.

The next example involves events where the source agent is "GOV". We first group them by geolocation and create a geo-map showing four types of "counts" - low, med, high, very high. Figure-9 shows the output.

Another example of visualization involves showing frequency distribution of events of type Material Conflict between India and Pakistan in the duration from January 3, 2018 and June 20, 2019. (Figure-10)

Related Applications

In this section we highlight some possible scenarios where mining the generated events can provide useful political or social understanding of different phenomena. As presented in the Section-, simplified analysis often reveals insights about the data, such as supporting information we already know or providing answers about unknown occurrences. We think the real-time and API datasets will continue to provide strong support to the research community for better understanding of political interactions. The framework used here can be applied to better understand non-political social phenomena (i.e,. Human Migration).

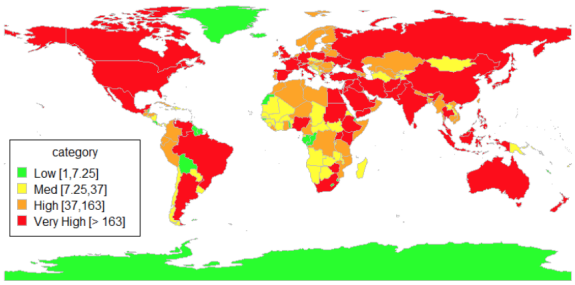

Figure 9: Discretized Frequency distribution of events with source agent = "GOV"

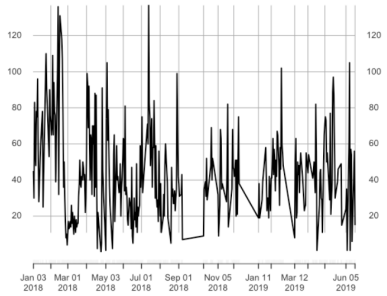

Figure 10: Events of type "Material Conflict" between India and Pakistan

To study human migration, we need a similar ontology-driven information extraction mechanism. We can rely on human annotators to provide us CAMEO-like dictionaries useful for capturing migration related events in news articles. We can semi-automate the process by using systems similar to the ones we use as the automatic actor detection framework. The key difference is we have to identify patterns involving verbs rather than named entities, which is a much harder problem to address. Once we have the necessary methods to identify patterns involving verbs, we can reuse this framework to apply to non-political phenomena.

Once we gather this information, either political or social, we can mine and extrapolate related events. For example, observing two entities and trying to predict if they are becoming prone to war can be within the scope of the data generated by this framework. Simple analytical study on previous wars and associated entities may reveal threshold values of associated parameters.

However, currently, war is a less common phenomena compared to political instability where the stability of a nation or organization is undermined and exploited. Identifying stability at a state or organizational level will be an appropriate scenario for the dataset and framework to identify. We will have human annotators to help us identify a timeline for the instability to occur (from start to end). We will do sequence analysis(Studer and Ritschard 2016) on the subset of the data related to before the event started (to identify the cause), during its progression, and the end of the event (to identify the sequence of phenomena that led to the result). Following sequences will be studied w.r.t the learned ones and classified to reflect potential future course of events.

Another key area of mining news articles is news credibility identification (aka Fake News Detection). Machine driven authentication for a statement or news article is still being explored as a research field. There are several parts in this validation mechanism eg., Fact Checking(Hassan et al. 2015), Relevancy analysis at article level(Chen, Johnson, and Yennie) etc. The data we collected will be used as a searchable knowledge-base for fact checking. As event coding works at a sentence level, a statement can be easily converted to a structured form for matching against other facts already collected.

Related Works

In this section we present the related works with respect to the contents of the paper. Distributed processing of large data has been addressed by (Solaimani et al. 2016), (Halterman et al. 2017a) where author address a static dataset and the process of generating events. Here, we are working on a real-time dataset and here we need to collect and distribute the data in real-time where aforementioned works only concentrate on processing the data. We are inspired by the scalability analysis found here (Solaimani et al. 2016) and incorporate that to design the system. Schordt(Schrodt 2012b) has pointed out the importance of having a real-time event coding framework and data distribution. Automated Political event coding has gone through several decades and several ontologies, datasets and tools are developed. We use CAMEO ontology here. Other related ontologies are WEIS and authors(Gerner et al. 2002) has provided a comparative study between those ontologies. Eck(Eck 2012) also present an analysis among different conflict data-sets.

Among the available datasets ICEWS(Boschee et al. 2018), Cline Center Dataset(Althaus and Shalmon 2017) are prominent. But they are not easily accessible and updated in-frequently. In terms of event coder, we found PETRARCH2 is the most flexible, easy-to-extend compared to others (i.e., BBN Accent) as reported by (Beieler 2016; Solaimani et al. 2017). We are also using a extended version of the event coder which uses Universal Dependency (McDonald et al. 2013) to better support towards foreign languages. Such type of extension was not possible to BBN Accent like proprietary software. Another dataset that matches with some types of events in CAMEO ontology would be the Global Terrorism Dataset(GTD)(gtd) which lists worldwide terrorist activities including different types of protest. The key aspect of the dataset is it is human annotated but doesn't link well with the orignal news source. We are currently working with this dataset to use it as a benchmark tool for the automatic event coder.

Future Works and Conclusion

In this paper, first we described real-time political events dataset and an infrustructure to sustain it. Second, we discussed about the data sharing process through rich query based API serving JSON data. Third, we discussed about multilingual event coding by updating POS based event

coder with universal dependency based one. Fourth, we presented summary of the data to give readers an insight about the gathered dataset. Finally, we discussed some future applications that can be built on top of the framework and the dataset.

In future, we will extend the system to capture other social and political phenomena. Currently we are working on a limited set of Spanish news sources. Once we increase that and add Arabic news sources, we need to study how the system performs in-terms of scalability. We will work on methods that can be used to semi-automate the ontology extension part. We will simplify the query mechanism to the API by incorporating more natural way of specifying parameters and query translation.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Nos. OAC-1931541, SBE-SMA-1539302, and SES-170012. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

References

[Althaus and Shalmon 2017] Althaus, Scott, J. B. J. F. C. B. P., and Shalmon, D. A. 2017. Cline Center Historical Phoenix Event Data. v.1.0.0.

[Beieler 2016] Beieler, J. 2016. Generating politically-relevant event data. *arXiv preprint arXiv:1609.06239*.

[Boschee et al. 2018] Boschee, E.; Lautenschlager, J.; O'Brien, S.; Shellman, S.; Starz, J.; and Ward, M. 2018. ICEWS Coded Event Data.

[Chen, Johnson, and Yennie] Chen, J. Y.; Johnson, J.; and Yennie, G. Rnns for stance detection between news articles.

[D'Orazio, Deng, and Shoemate 2018] D'Orazio, V.; Deng, M.; and Shoemate, M. 2018. Tworavens for event data. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, 394–401. IEEE.

[D'Ignazio et al. 2014] D'Ignazio, C.; Bhargava, R.; Zuckerman, E.; and Beck, L. 2014. Cliff-clavin: Determining geographic focus for news. *NewsKDD: Data Science for News Publishing, at KDD* 2014.

[Eck 2012] Eck, K. 2012. In data we trust? a comparison of ucdp ged and acled conflict events datasets. *Cooperation and Conflict* 47(1):124–141.

[Fellbaum 1998] Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

[Gerner et al. 2002] Gerner, D. J.; Schrodt, P. A.; Yilmaz, O.; and Abu-Jabr, R. 2002. Conflict and mediation event observations (cameo): A new event data framework for the analysis of foreign policy interactions. *International Studies Association, New Orleans*.

[Goldstein 1992] Goldstein, J. S. 1992. A conflict-cooperation scale for weis events data. *The Journal of Conflict Resolution* 36(2):369–385.

[gtd] Global terrorism dataset.

[Halterman et al. 2017a] Halterman, A.; Irvine, J.; Landis, M.; Jalla, P.; Liang, Y.; Grant, C.; and Solaimani, M. 2017a. Adaptive scalable pipelines for political event data generation. In *2017 IEEE International Conference on Big Data (Big Data)*, 2879–2883.

[Halterman et al. 2017b] Halterman, A.; Irvine, J.; Landis, M.; Jalla, P.; Liang, Y.; Grant, C.; and Solaimani, M. 2017b. Adaptive scalable pipelines for political event data generation. In *2017 IEEE International Conference on Big Data (Big Data)*, 2879–2883. IEEE.

[Hassan et al. 2015] Hassan, N.; Adair, B.; Hamilton, J. T.; Li, C.; Tremayne, M.; Yang, J.; and Yu, C. 2015. The quest to automate fact-checking. *world*.

[Khan and McLeod 2000] Khan, L., and McLeod, D. 2000. Effective retrieval of audio information from annotated text using ontologies. In *Proceedings of the International Workshop on Multimedia Data Mining, MDM/KDD'2000, August 20th, 2000, Boston, MA, USA*, 37–45.

[Kim et al. 2019] Kim, H.; D'Orazio, V.; Brandt, P.; Looper, J.; Salam, S.; Khan, L.; and Shoemate, M. 2019. Utdeventdata: An r package to access political event data. *Journal of Open Source Software* 4(36):1322.

[Lu 2019] Lu, J. 2019. Universal dependency based petrarch, language-agnostic political event coding using universal dependencies.

[McDonald et al. 2013] McDonald, R.; Nivre, J.; Quirmbach-Brundage, Y.; Goldberg, Y.; Das, D.; Ganchev, K.; Hall, K.; Petrov, S.; Zhang, H.; Täckström, O.; et al. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, 92–97.

[Miller, Vandome, and McBrewster 2009] Miller, F. P.; Vandome, A. F.; and McBrewster, J. 2009. *Levenshtein Distance: Information Theory, Computer Science, String (Computer Science), String Metric, Damerau?Levenshtein Distance, Spell Checker, Hamming Distance*. Alpha Press.

[Müllner 2011] Müllner, D. 2011. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*.

[Muniz, Chalub, and Rademaker 2017] Muniz, H.; Chalub, F.; and Rademaker, A. 2017. Cl-conllu: dependências universais em common lisp. https://sites.google.com/view/tilic2017/.

[Navigli and Ponzetto 2012] Navigli, R., and Ponzetto, S. P. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence* 193:217–250.

[Norris, Schrodt, and Beieler 2017] Norris, C.; Schrodt, P.; and Beieler, J. 2017. PETRARCH2: Another event coding program. *The Journal of Open Source Software* 2(9).

[OEDA 2018] OEDA. 2018. List of news sources in spanish.

[Osorio et al. 2019] Osorio, J.; Pavon, V.; Salam, S.; Holmes, J.; Brandt, P. T.; and Khan, L. 2019. Translating cameo verbs for automated coding of event data. *International Interactions* 45(6):1049–1064.

[Salam et al. 2020] Salam, S.; Khan, D. L.; Brandt, D. P.; D'Orazio, D. V.; and Holmes, D. J. 2020. CAMEO Ontology Based Real-time Political Event Data.

[Schrodt 2012a] Schrodt, P. A. 2012a. Cameo: Conflict and mediation event observations event and actor codebook. *Pennsylvania State University*.

[Schrodt 2012b] Schrodt, P. A. 2012b. Precedents, Progress, and Prospects in Political Event Data Article in International Interactions.

[Skorupa Parolin et al. 2019] Skorupa Parolin, E.; Salam, S.; Khan, L.; Brandt, P.; and Holmes, J. 2019. Automated verbal-pattern extraction from political news articles using cameo event coding ontology. In *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, 258–266.

[Solaimani et al. 2016] Solaimani, M.; Gopalan, R.; Khan, L.; Brandt, P. T.; and Thuraisingham, B. 2016. Spark-based political event coding. In *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*, 14–23.

[Solaimani et al. 2017] Solaimani, M.; Salam, S.; Khan, L.; Brandt, P. T.; and D'Orazio, V. 2017. Repair: Recommend political actors in real-time from news websites. In *2017 IEEE International Conference on Big Data (Big Data)*, 1333–1340.

[Steinberger et al. 2013] Steinberger, R.; Pouliquen, B.; Kabadjov, M.; and Van der Goot, E. 2013. Jrc-names: A freely available, highly multilingual named entity resource. *arXiv preprint arXiv:1309.6162*.

[Stewart et al. 2015] Stewart, C. A.; Cockerill, T. M.; Foster, I. T.; Hancock, D. Y.; Merchant, N.; Skidmore, E.; Stanzione, D.; Taylor, J.; Tuecke, S.; Turner, G. W.; et al. 2015. Jetstream: a self-provisioned, scalable science and engineering cloud environment. In *XSEDE*, 29–1.

[Studer and Ritschard 2016] Studer, M., and Ritschard, G. 2016. What matters in differences between life trajectories: A comparative review of sequence dissimilarity measures. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 179(2):481–511.

[whi 2016] 2016. List of urls for english news feeds.

[Wikipedia contributors 2019] Wikipedia contributors. 2019. Elbow method (clustering) — Wikipedia, the free encyclopedia. [Online; accessed 20-August-2019].