EMBERS AutoGSR:
Automated Coding of Civil Unrest Events

Parang Saraf
Discovery Analytics Center
Department of Computer Science
Virginia Tech
parang@cs.vt.edu

Naren Ramakrishnan
Discovery Analytics Center
Department of Computer Science
Virginia Tech
naren@cs.vt.edu

ABSTRACT
We describe the EMBERS AutoGSR system that conducts automated coding of civil unrest events from news articles published in multiple languages. The nuts and bolts of the AutoGSR system constitute an ecosystem of filtering, ranking, and recommendation models to determine if an article reports a civil unrest event and, if so, to proceed to identify and encode specific characteristics of the civil unrest event such as the when, where, who, and why of the protest. AutoGSR is a deployed system for the past 6 months continually processing data 24x7 in languages such as Spanish, Portuguese, English and encoding civil unrest events in 10 countries of Latin America: Argentina, Brazil, Chile, Colombia, Ecuador, El Salvador, Mexico, Paraguay, Uruguay, and Venezuela. We demonstrate the superiority of AutoGSR over both manual approaches and other state-of-the-art encoding systems for civil unrest.

Keywords
event extraction, event encoding, text mining

1. INTRODUCTION
The computational modeling and interpretation of societal events has been a holy grail in social science research. Beginning in the 1980s, there have been significant efforts in computational analysis of societal events supported by government programs such as DARPA’s ICEWS (Integrated Conflict Early Warning System) and CIA’s PITF (Political Instability Task Force). Projects of similar (and more ambitious) scope continues to this day, and offer greater specificity, both spatially and temporally, into modeling global events.

We are part of the EMBERS consortium [13] that aims to forecast civil unrest phenomena (protests, strikes, and ‘occupy’ events) in multiple countries of Latin America, specifically Argentina, Brazil, Chile, Colombia, Ecuador, El Salvador, Mexico, Paraguay, Uruguay, and Venezuela. In our earlier KDD 2014 paper [13] we demonstrated how we can use open source indicators such as news, blogs, Twitter, food prices, and economic data, to forecast civil unrest events. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org

EMBERS forecasts have been evaluated by a third party (MITRE) wherein human analysts prepared a ground truth dataset (called the GSR, or ‘Gold Standard Report’) of reported protests in newspapers of record. The GSR is compared against EMBERS forecasts using metrics introduced in [13].

As the EMBERS project matures, we realized that we must pay attention to not just forecasting events but also to coding ongoing events, i.e., the process of constructing the GSR on a regular basis. For instance, see Fig. 2. Such coded data serves two uses in EMBERS: to help evaluate EMBERS forecasts, and to support the regular re-training and tuning of the machine learning models. Accordingly, we launched a parallel effort, referred to as the EMBERS AutoGSR, that conducts automated coding of civil unrest events from news articles published in multiple languages. Like EMBERS, the AutoGSR is also a deployed system continually processing data 24x7 in languages such as Spanish, Portuguese, English. Our key contributions can be summarized as follows:

1. To the best of our knowledge, the AutoGSR is the only/first system to be able to automatically encode civil unrest events across 10 countries in languages local to these countries. This gives a significant advantage over systems that are manual or systems that are automatic but restricted to English.

2. The nuts and bolts of the AutoGSR system constitute an ecosystem of filtering, ranking, and recommendation models to determine if an article reports a civil unrest event and, if so, to proceed to identify and encode specific characteristics of a civil unrest event such as the when, where, who, and why of the protest. We present an exhaustive evaluation of the performance of AutoGSR using metrics in the large (e.g., does the system track ongoing happenings in countries of interest?) as well as metrics in the small (e.g., does the system identify specific events of interest?).

3. AutoGSR is meant to be used in both a fully automated and an analyst assisted mode. Through detailed analysis of hours logged in both the manual process and in the AutoGSR system, we quantify the performance gains of our approach.

2. RELATED WORK
The challenges associated with a system like AutoGSR can be broadly classified in two categories: event encoders and event databases. While the event encoders are not freely available, the event databases constructed around them are more available.
Table 1: Sample erroneous encodings by ICEWS and GDELT.

<table>
<thead>
<tr>
<th>Source</th>
<th>ID</th>
<th>Representative Paragraph</th>
<th>Reason for Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDELT</td>
<td>299144197</td>
<td>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.</td>
<td>Presence of the word ‘demonstrate’ results in a false positive.</td>
</tr>
<tr>
<td>GDELT</td>
<td>256666928 and 256814375</td>
<td>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.</td>
<td>No Duplicate Detection. Two events extracted from two articles covering the same story but published on different days.</td>
</tr>
<tr>
<td>ICEWS</td>
<td>23909784</td>
<td>Since its inception, the Islamic State group has demonstrated the firmness of its structure and the strength of its organizational composition.</td>
<td>Presence of word ‘demonstrate’ results in a false positive.</td>
</tr>
<tr>
<td>ICEWS</td>
<td>19873295</td>
<td>Capriles has called off a march by his supporters in Caracas, saying that his rivals were plotting to “infiltrate” the rally to trigger violence.</td>
<td>The protest was called off.</td>
</tr>
</tbody>
</table>

2.1 Event Encoders

Hogenboom et. al.[3] provides an overview of different extraction methodologies used by the current state-of-the-art systems. The methodologies used here include statistical as well as linguistic and lexicographical approaches for event extraction. TABARI (Textual Analysis by Augmented Replacement Instructions) [15] and BBN’s SERIF (Statistical Entity and Relation Information Finder) [1, 14] are two state-of-the-art event encoders. These encoders not only extract events but also encode them via a mapping to an event taxonomy. Such mappings add structure to the extracted event thereby making it feasible to perform systematic studies. CAMEO [16] is one such widely used event taxonomy and is used by both TABARI and SERIF.

TABARI is one of the earlier open source event extraction systems that uses sparse parsing to recognize patterns in text. These patterns are hand coded and identify three types of information: actors, verbs, and actions. For a given text, only a few initial sentences are used for event extraction, to support high throughput applications. Several improved versions of TABARI have been proposed. JABARI is one such system, which is a Java implementation of TABARI and uses a few advanced NLP techniques in addition to pattern matching. More recently, PETRARCH has emerged as the successor of TABARI. Instead of conducting a pattern based extraction, PETRARCH uses the full parsed Penn TreeBank as input to perform a parser-based encoding.

BBN’s SERIF is another state-of-the-art event encoder that uses a series of NLP components to capture representations of type ‘who did what to whom’ in article text. The encoder works at both the sentence and document level and is able to identify and resolve entities between sentences. Once the entities are resolved, the encoder detects and characterizes the relationship between entities. Finally the encoder maps these relationships to the CAMEO taxonomy using an externally provided list of actor dictionaries and event patterns.

2.2 Event Databases

The origins of automated event databases can be attributed primarily to political scientists and intelligence agencies who over the years have envisioned systems that can perform large scale encoding of events by mining million of news articles. ICEWS (Integrated Crisis Early Warning System) [11] and GDELT (Global Database of Events, Language and Tone)[6]
are two such systems that analyze hundreds of news sources from all over the world in order to generate a database of events. ICEWS, which began in 2007, is a DARPA funded project that focuses primarily on monitoring, accessing and forecasting events of interests for military commanders. Internally, ICEWS employs TABARI and SERIF to encode news articles. ICEWS focuses primarily on generating high quality, reliable events and uses several mechanisms to filter the raw stream of reported stories into a unique stream of events. Events are encoded in accordance to the CAMEO taxonomy.

GDELT, on the other hand focuses on capturing an extensive set of events both in terms of categories and geographical spread. By design, the goal of GDELT is to collect a large number of events without worrying about false positives. Internally, it uses an enhanced version of TABARI and maps events to the CAMEO taxonomy.

Although both these systems are considered state-of-the-art, in our experiments we have found that they perform poorly in comparison to manually generated ground truth data. See Table 1 for examples of erroneous encodings in ICEWS and GDELT. This is primarily because of the fully automated event generation process that yields false positives. Hence, there is a need for a semi-automated system that can generate validated event encodings with minimal human effort. Table 2 contrasts ICEWS and GDELT against our AutoGSR system.

### 3. SYSTEM ARCHITECTURE

EMBERS AutoGSR is a web based system that generates validated civil unrest events extracted from news articles. The system architecture is shown in Fig. 3.

#### 3.1 Data Sources

News articles are collected every day from three data sources: 1) Site specific Google Search 2) RSS feeds subscribed to individual news websites, and 3) News databases. Articles are filtered if they contain a protest related keyword. With the help of subject matter experts, we created a list of protest keywords for each language.

#### 3.2 Data Processing

This stage of EMBERS AutoGSR is responsible for running and managing the sophisticated models ecosystem, along with performing standard tasks like data cleaning and enrichment.

##### 3.2.1 Fetch and Clean

The fetch and clean component fetches the article from the original web source and performs a boilerplate removal in order to extract the full article text. On this extracted full text, keyword based search is performed once again to make sure that at least one protest related keyword is present in the text. This component also fetches SEO meta tags, if present for each article. (These meta tags are added by news websites in order to improve their search ranking and includes several key information about the article such as publish date, keywords, summary, and description.)

##### 3.2.2 Data Enrichment

This component enriches news articles by performing various kinds of linguistic processing. The enrichment comprises named entity extraction, lemmatization, location identification, geo-reference resolution, and translation of article text to English. This enriched information is used by models in the models ecosystem.

Words or noun phrases identified as a location by the named entity resolver are passed through a geo reference resolver to resolve ambiguity, if any. Many a times, locations are referred by their alternate name(s), for example: LA for City of Los Angeles or America/US/USA for The United States of America. Alternatively, several times, famous landmarks like Times Square, White House, etc are used to denote a location. The geo-reference resolver goes through a world gazetteer\(^1\) to resolve these phrases and maps them to the official name.

Please note that articles are translated into english only to enhance the readability of an end user. The translated text is not used by any of the models. The models are designed to work with native language.

##### 3.2.3 Models Ecosystem

This component, explained in much detail in a subsequent section, applies several machine learning models on news articles. A first set of models classify an incoming article into protest or non protest. If the article is classified as a protest article, then it passes through a second set of models that recommend event encoding(s) for the article.

---

\(^1\) GeoNames: http://www.geonames.org/
Filtering Models. These models are based on hand coded rules that perform a hard classification into protest or non-protest. The models are applied in a series, where if the article fails one of the models, the article is classified as a non-protest article. When an article successfully passes all the filtering models, it is processed through the subsequent ranking and recommendation models.

Ranking Models. These models work independently of each other and assign a probability score for classifying the article as a protest article. The individual probability scores generated by each model are fed to an ensembler which generates a representative ensemble probability score for the article. The accuracy of individual models in the past is also used as an input by the ensembler.

Recommendation Models. The recommendation models assume that the article is a protest article and based on this assumption extracts & recommends both full event encoding(s) and individual encoding components. Elaborating on what was mentioned earlier, a full event encoding comprises of the following components: 1) protest location, 2) protest date, 3) participating population group, 4) reason for protest, 5) violent or peaceful protest, and 6) protest reported date. These models work in tandem to generate the full event encoding. If there are multiple models generating recommendations for a particular encoding component then an ensembler is used to determine a representative value.

3.3 Database
The output of the data processing component containing clean, enriched news articles along with the individual and ensembled values generated by the model ecosystem is stored in a database. This database acts as the primary data source for the AutoGSR Web Interface which displays news articles along with the associated data. For each article, the database also stores validated encoding results generated by the manual validation process.

3.4 Web Interface
The web interface shown in Fig. 4 displays enriched news articles; output from the models ecosystem; and controls to validate recommended encodings. More specifically the interface displays the following (the numerical labeling of the components in the image corresponds with the list numbers below):

1. Allows the user to specify the filtering criteria for displaying news articles. The system also allows a user to specify filtering criteria for displaying news articles. An important criterion is the cutoff probability with values between 0 and 1 and is used to classify an article as protest or non-protest. If the representative ensembled probability of an article, generated by the ranking models ensembler, is greater than this cutoff probability, then the article is classified as a protest article; otherwise it gets labeled as a non-protest article.

Using this control, the user can also tune the precision and recall of the system.

2. The articles satisfying the filtering criteria are clustered in real time to generate news clusters. Each news cluster brings together related stories. For each cluster, a representative label is also generated. The real time clustering is performed using the Lingo3G clustering suite\(^2\). A summary text is also generated for each article in the cluster, using the description meta tag.

3. For each article, the detailed view shows the full article text, article image and translated english text along with recommended event encoding and the output of the models ecosystem as described next.

4. Event encodings are recommended in following three ways – a) Ensembled Recommendations: These full encoding recommendations are generated by putting together the recommendations for individual encoding components as generated by the recommendation models from the ecosystem. In the case of automated event extraction, these recommended encodings are considered as the extracted encoding for the article.

b) Clustering Based Recommendations: Full encoding recommendations are generated using related articles with validated encodings in the dynamically generated news clusters. These recommendations assume that similar/related articles will result in almost similar encodings.

c) Individual Recommendations: The individual components in the encoding validation form show component specific recommendations.

5. All sentences used by the system to generate recommendations are highlighted for the user's reference. There is an option to toggle the reading view, where only the highlighted sentences of the article are shown. The ‘Highlighted Text’ view assumes that, in the ideal scenario, the highlighted sentences will provide complete information about the article. If the user doesn’t agree with the highlighted sentence or the kind of information identified by it, then he can click on that particular sentence and modify the information type through the popover. This active feedback helps the system to learn.

3.5 Encoding Validation
For each article classified as a protest article, recommended encodings are validated manually by subject matter experts. During the validation process, the experts can either accept the recommended encoding or modify it. The validation process is performed using the validation controls of the interface. Encodings for each article are validated by at least two analysts. If the analysts agree on the encoding then that encoding is considered as the final encoding. However, in case of a conflict, the final encoding is decided by a quality control analyst.

3.6 Event Generation
Based on the validated records, a final set of events is generated by performing duplicate detection. As there can be multiple articles reporting the same protest, duplicate detection is performed. During duplicate detection, unique event encoding tuples comprising <<protest location, protest date, protest reason, participating population group, violence/peaceful>> are identified and resolved.

4. MODELS ECOSYSTEM
In civil unrest encoding, the ratio of positive to negative articles is highly skewed towards negative articles. Based on

\(^2\) Lingo3G: https://carrotsearch.com/
our experience, given a set of articles containing at least one protest keyword, this ratio is somewhere around 1:8. Due to this skewness, an analyst has to typically skim through a huge number of negative articles just to identify a few protest articles. What makes this problem even more challenging is the inherent inability to characterize negative articles. While we know what constitutes a protest article, there is no simple way to describe a non-protest article. Given these challenges, the aim of the models ecosystem is two-fold: 1) reduce the number of negative articles while minimizing the loss of positive articles, and 2) generate event encoding recommendations for positive articles.

Both of these are non-trivial problems and therefore the solution requires an ensemble of models, each with its own superiorities, focus area, and performance. The models as shown in Table 3 and categorized into three compartments: 1) Filtering Models, 2) Ranking Models, and 3) Recommendation Models.

4.1 Filtering Models

These are rules-based models that perform hard classification of news articles into protest and non-protest. If the article fails even one of these models, then the article is classified as a non-protest article.

4.1.1 Sub Domain Based Filtering

Many of the sub-domains, for example entertainment, technology, sports, etc., are tagged as non-relevant. If an article is published in any of these sub-domains, then the article is not considered relevant and gets classified as non-protest. In this context, a sub-domain worth mentioning is sports. Sports articles use a lot of protest keywords like attack, surrounded, etc., to describe competition between two teams/players.

4.1.2 URL Based Filtering

There are several URL structures that list multiple articles on a single page. For example, URLs listing top stories of the day,

3 http://www.clarin.com/politica/

or URLs listing stories by topic,

4 http://www.clarin.com/tema/manifestaciones.html

or URLs corresponding to search terms,

5 http://www.clarin.com/buscador?q=protesta

are considered irrelevant for this task, and any article with URL matching one of these types is discarded.

4.1.3 Negative Keywords Based Filtering

For many of the protest keywords, there exist words, called Negative Keywords which when used in the vicinity of a protest word can completely alter the meaning. A few such negative keywords are described in Table 4. If any such negative keyword is present in the vicinity of the protest keyword in the same sentence, then such matches are not considered as a positive keyword match and are ignored. After ignoring all such false matches, if the article still contains any other protest keyword, only then the article is allowed to pass. Otherwise, the article is reported as a non-protest article.

4.2 Ranking Models

Each of these models assigns a probability score of classi-

3 http://www.clarin.com/politica/

http://www.clarin.com/tema/manifestaciones.html

http://www.clarin.com/buscador?q=protesta
### Filtering Models

| These are rules based models that classify incoming news articles into protest and non protest with a 0 or 1 certainty |
| 1. Sub Domain Based Filtering  
2. URL Based Filtering  
3. Negative Keywords Based Filtering |

### Ranking Models

| These models use standard machine learning algorithms on different components of an incoming article for classification. |
| 1. Labeled Unlabeled Text Classifier  
2. Entity Based Classifier  
3. Distributed Representation Classifier  
4. MetaTags Based Classifier |

### Recommendations Models

| These models assume that the incoming article is a protest article and tries to recommend complete or partial encoding(s) for the article |
| Protest Identification Models:  
1. Geo Location Recommender  
2. Temporal Recommender  
Protest Characterization Models:  
3. Entity Based Naïve Bayes Recommender  
4. MetaTags Based Recommender  
Usability Models:  
5. Clustering Based Recommender |

## Approach

**Approach:** Articles are passed through these models sequentially. If any of these models classifies the article as non protest then the article is labeled as a non protest article in the interface.

**Approach:** Each of these models assigns individual probabilities to an incoming article. An article’s final probability is calculated using ‘model ensemble’ approach. In the interface user can specify a cutoff probability score. Articles that have probability greater than the cutoff will appear as protest articles.

**Approach:** These recommendations appear in the interface for each article. The recommendations are generated for both full encodings and individual encoding components.

### Example Negative Keywords

Table 4: Example Negative Keywords

<table>
<thead>
<tr>
<th>Protest Keyword</th>
<th>Negative Keyword Phrase</th>
<th>Altered Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>marcha</td>
<td>ponar en marcha</td>
<td>to start; set in motion</td>
</tr>
<tr>
<td>protesta</td>
<td>tomar protesta</td>
<td>to swear in (public official)</td>
</tr>
<tr>
<td>protesta</td>
<td>rendir protesta</td>
<td>to swear in (public official)</td>
</tr>
</tbody>
</table>

### MetaTags based Classifier

In the literature, much of effort in classification of web pages or news articles has focused only on full article text. Almost little to no work exists in the domain of using HTML meta tags as input. HTML meta tags provide a wide variety of structured information about a web page, which is primarily used by search engine crawlers. In our scenario, tags of much importance are the ones that are used to generate rich text snippets. Rich text snippet is a short summary description of a web page that is shown when the page appears in search engine results and/or social media websites. Generally, this short description is manually provided by the author of the news article to succinctly describe the article. Google uses meta tags description and title, along with several other latent parameters to generate snippets. Facebook has created the Open Graph Protocol that describes guidelines for meta tags. These guidelines allow content creators to control the display of their content on social networking websites.
Similarly, Twitter which supports the open graph protocol, also has its own guidelines for content sharing.

<table>
<thead>
<tr>
<th>Targeted For</th>
<th>Tag Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Engines</td>
<td>title, description</td>
</tr>
<tr>
<td>Facebook</td>
<td>og:title, og:description</td>
</tr>
<tr>
<td>Twitter</td>
<td>twitter:title, twitter:description</td>
</tr>
</tbody>
</table>

Table 5 lists the various standard meta tags that we extract information from. In this model, we work on the text extracted from only these tags. We train a vanilla SVM classifier [2] with a linear kernel that has been shown to work exceptionally well with text classification tasks [4].

### 4.2.5 Ensembler

The goal of the ensembler is to take output probabilities from the individual ranking models and fuse them together to generate a final representative probability of the article. The representative probability score should provide a better classification accuracy than any of the models individually. We generate the representative probability score by performing a weighted average of the individual outputs, where weights correspond to the accuracy of each model:  
\[
P(l_p) = \sum_{i=1}^{M} c_i o_i \frac{\sum_{m=1}^{M} c_m}{\sum_{i=1}^{M} c_i}
\]

where, \(P(l_p)\) is the probability of classifying the article as protest article, \(c_i\) is the accuracy and \(o_i\) is the output of model \(i\). The accuracy of a model is derived empirically based on manual validation.

### 4.3 Recommendation Models

The goal of the recommendation models is to generate partial or complete encoding suggestions for a given article. Broadly speaking these models can be grouped in three categories: 1) Protest Identification, 2) Protest Characterization, and 3) Usability. The **Protest Identification** models aim to identify when and where a protest happened: Geo Location Recommender along with Temporal Recommender identify city and date of protest. Once a protest has been identified, the **Protest Characterization** models work on characterizing a protest event by identifying the reason behind protest and the participating population group. Entity Based Naïve Bayes Recommender and MetaTags based recommender are two such models. The output from these models is passed through an **enssembler**, that combines these individual suggestions into a complete encoding. The interface shows both complete as ensembler confidence score of that date.

#### 4.3.1 Geo-location Recommender

This is a protest identification model, and aims to identify protest location(s) from a news article. The model begins by geo resolving the ‘location’ named entities found in the article. Geo resolving involves mapping location named entities to cities. These location entities can correspond to local points of interest, alternative name for the city or the official city name. The points of interest pose an interesting challenge as they tend to match to multiple cities. For example: Main St. can be found in almost every US city. The goal of the geo resolution is to map these location entities to a minimum number of cities with an assumption that generally a given news article refers only to a focused set of locations. We use the GeoNames database and perform the resolution in a top down fashion. First, all the cities are identified and then the ambiguous points of interest are mapped to these cities. The cities are assigned a confidence rankings based on the total number of mapping location entities. The case where the same city name is present in multiple states is resolved by selecting the most probable city based on past validated encodings.

Frequently, articles report a statewide or a nationwide protest. Such cases are identified by searching for various language specific forms of the keywords: statewide and nationwide, in the article. If such words are found, then we add the top level state/country to our list of location recommendations.

#### 4.3.2 Temporal Recommender

This is another protest identification model that focuses on identifying protest date(s) from a news article. Article publish date is used as a date of reference and is identified either by the publish date meta tag or is inferred by the article fetch date. The reference date allows the model to resolve words like yesterday or last Tuesday. The temporal keywords are searched in following places: title, description meta tag and the sentences containing protest keyword(s). Temporal keywords found in title and description carry more confidence score as compared to the ones found in article text. Also, multiple mentions of the same date, increases the confidence score of that date.

#### 4.3.3 Entity Based Recommender

This is a protest characterization model that identifies the salient features of a protest such as the reason of protest, participating population group and whether the protest was violent or peaceful. In context of AutoGSR, the reason behind a protest can be defined in only following six categories: 1) Employment and Wages, 2) Housing, 3) Energy and Resources, 4) Other Economic Policies, 5) Other Government Policies, and 6) Other. Similarly, the participating population group can be one among the following eleven: Agriculture, Business, Education, Ethnic, General Population, Labor, Legal, Media, Medical, Refugees and Religious. This model employs a multi class Naïve Bayes model trained on named entities extracted from article text and past protest encodings to generate probability score for event Type and population group categories.

#### 4.3.4 MetaTags Based Recommender

This model is also a protest characterization model, and works very similar to the Entity Based Recommender in principle, but uses only the entities extracted from the title and the description MetaTag to determine event type and population group categories.

#### 4.3.5 Ensembler

The task of the ensembler is to take the probabilistic outputs of the protest identification and characterizations models and generate complete encoding recommendations. The complete encoding recommendation includes location, date, event type and population group and is generated by finding the most probable individual recommendations in
each category. Top two such complete recommendations are shown in the interface.

4.3.6 Clustering Based Recommender

This is a usability model that aims to increase the ease of using the system in two ways: first, by clustering similar news articles together and second by generating full tuple encoding recommendations based on validated encodings of other articles in the article cluster. Article clusters are generated in real time using the Lingo3G document clustering engine[12]. These clusters allow an analyst to work on related news articles together. Encoding recommendations are generated for un-validated documents using the validated encodings of other documents in the cluster. These recommendations are generated assuming that articles in the clusters are related and hence will have similar encoding. Please note that these recommendations are in addition to the recommendations generated by the ensembler.

5. EVALUATION RESULTS

As stated earlier, the goal of the EMBERS AutoGSR is to develop reliable ground truth civil unrest events while minimizing the manual effort required to do so. With this goal in mind, we evaluate our system alongside four aspects:

1. What is the reduction in number of articles realized after each step of the models ecosystem? (Section 5.1)
2. How does AutoGSR compare w.r.t. manually generated ground truth data? (Section 5.2)
3. What is the reduction in manual effort afforded by AutoGSR? (Section 5.3)
4. How does AutoGSR compare with state-of-the-art systems like ICEWS and GDELT? (Section 5.4)

The evaluation is performed using a manually generated list of civil unrest events for the same period, countries and news sources. These events are hand coded by a team of 15-18 political scientists and subject matter experts working for an independent agency (MITRE). As stated earlier, we refer to the MITRE organized set of ground truth events as GSR, to distinguish them from our AutoGSR system. We focus here primarily on 3 recent months: October, November and December, 2015 for the 10 Latin American countries mentioned earlier, viz. Argentina, Brazil, Chile, Colombia, Ecuador, El Salvador, Mexico, Paraguay, Uruguay, and Venezuela. The news articles are primarily reported in two languages: Spanish and Portuguese, with some English sources as well.

5.1 Reduction in number of articles

Here we present the overall reduction in number of articles for both protest and non protest articles after each stage of the AutoGSR pipeline for both Spanish (ES) and Portuguese (PT), fig. 5. For Spanish, the reduction in non protest articles after filtering models is 60% and after ranking models is 64%, thereby leading to a net reduction of 86%. Similarly, for Portuguese, the reduction in non protest articles after filtering models is 34% and after ranking models is 60%, thereby leading to a net reduction of 74%.

5.2 Quality and Coverage Evaluation

In order to determine quality, we need to identify how similar GSR and AutoGSR encoding extractions are for a given civil unrest event. As mentioned before, an event encoding consists of the following four entities – protest location (city level), protest date, reason of protest along with violence identification (event type) and protesting population group.

We define Quality Score (QS) on a 4 point scale where each point correspond to how similar each of the encoding entity is as explained below:

\[ \text{QualityScore}(QS) = DS + LS + ES + PS \]

where DS, LS, ES and PS denote the date score, location score, event type score and population score respectively. Each of these scores are defined as:

1. **Date score (DS)**: $DS = 1 - \min(|\text{GSR}\text{eventdate} - \text{AutoGSR}\text{eventdate}|, 7)/7$
2. **Location score (LS)**: $LS = 0.33 + 0.66(1 - \min(\text{dist}, 300)/300)$
3. **Event type score (ES)**: $ES = \frac{1}{3}e_1 + \frac{1}{3}e_2 + \frac{1}{3}e_3$
4. **Population score (PS)**: $PS = 0$ if they don’t match and 1 if they match.

Population score (PS) is simply a binary (0/1) score denoting whether we identified the correct population group or not. Finally, note that QS is designed to take values in the range [0, 4].

Note that the above mentioned Quality Score is generated only between a pair of event encodings for gsr and auto-gsr. For a given month, there can be thousands of such pairs and hence we need to find the most optimal mapping pairs. We resolve this issue by first constructing a bipartite graph between GSR and AutoGSR events where edge weights correspond to Quality Score between each pair. Then we construct a maximum weighted bipartite matching and consider this the most optimal mapping between GSR and AutoGSR events.

Table 6 shows the Average Quality Score, Precision and Recall for each of the ten countries for three months: Oct
Table 6: Quality and Coverage Evaluation for AutoGSR vs Manual GSR*.

<table>
<thead>
<tr>
<th>Country</th>
<th>Quality Score</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Oct</td>
<td>Nov</td>
<td>Dec</td>
</tr>
<tr>
<td>Argentina</td>
<td>3.24</td>
<td>3.21</td>
<td>-</td>
</tr>
<tr>
<td>Brazil</td>
<td>3.24</td>
<td>3.58</td>
<td>-</td>
</tr>
<tr>
<td>Chile</td>
<td>3.24</td>
<td>3.49</td>
<td>3.35</td>
</tr>
<tr>
<td>Colombia</td>
<td>3.17</td>
<td>3.24</td>
<td>2.71</td>
</tr>
<tr>
<td>Ecuador</td>
<td>3.99</td>
<td>3.20</td>
<td>3.20</td>
</tr>
<tr>
<td>El Salvador</td>
<td>3.33</td>
<td>3.09</td>
<td>3.10</td>
</tr>
<tr>
<td>Mexico</td>
<td>3.13</td>
<td>3.36</td>
<td>3.16</td>
</tr>
<tr>
<td>Paraguay</td>
<td>3.60</td>
<td>3.42</td>
<td>3.39</td>
</tr>
<tr>
<td>Uruguay</td>
<td>3.17</td>
<td>3.01</td>
<td>3.12</td>
</tr>
<tr>
<td>Venezuela</td>
<td>3.48</td>
<td>3.39</td>
<td>-</td>
</tr>
</tbody>
</table>

*In October no manual GSR was generated for Argentina, Brazil and Venezuela. Hence, we are unable to evaluate AutoGSR’s performance.

Table 7: Daily Time Series Correlation Comparison of ICEWS, GDELT and AutoGSR with Manual GSR.

<table>
<thead>
<tr>
<th>Country</th>
<th>ICEWS vs. GSR</th>
<th>GDELT vs. GSR</th>
<th>AutoGSR vs. GSR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argentina</td>
<td>0.13</td>
<td>0.26</td>
<td>0.49</td>
</tr>
<tr>
<td>Brazil</td>
<td>0.35</td>
<td>0.29</td>
<td>0.70</td>
</tr>
<tr>
<td>Chile</td>
<td>0.21</td>
<td>0.28</td>
<td>0.51</td>
</tr>
<tr>
<td>Colombia</td>
<td>0.33</td>
<td>0.23</td>
<td>0.48</td>
</tr>
<tr>
<td>Ecuador</td>
<td>0.50</td>
<td>0.19</td>
<td>0.73</td>
</tr>
<tr>
<td>El Salvador</td>
<td>nan</td>
<td>0.06</td>
<td>0.57</td>
</tr>
<tr>
<td>Mexico</td>
<td>-0.09</td>
<td>0.35</td>
<td>0.26</td>
</tr>
<tr>
<td>Paraguay</td>
<td>-0.12</td>
<td>0.10</td>
<td>0.86</td>
</tr>
<tr>
<td>Uruguay</td>
<td>-0.11</td>
<td>0.00</td>
<td>0.41</td>
</tr>
<tr>
<td>Venezuela</td>
<td>0.08</td>
<td>-0.08</td>
<td>0.68</td>
</tr>
</tbody>
</table>

5.3 Reduction in Manual Effort

We evaluate both objectively and subjectively the reduction in manual effort. The objective evaluation is performed by comparing total resource hours required to generate GSR and AutoGSR, while subjective evaluation is done by surveying MITRE’s analysts.

Tables 8 and 9 list the number of hours used by analysts to generate GSR and AutoGSR events. Table 10 shows minimum and maximum number of resource hours required per week by GSR and AutoGSR teams. The table also lists reduction figures. AutoGSR achieves an average reduction of 71-72% in manual effort.

In order to develop intuition behind this reduction in manual efforts, we surveyed the analysts from the MITRE team.

5.4 Comparison with ICEWS and GDELT

Finally, we evaluate AutoGSR’s performance against the current state of the art systems: ICEWS and GDELT. For each of these systems, for the given countries and time period, we compare the daily events counts. Table 7 shows the Pearson correlation values between daily time series for each of the ten countries for ICEWS-GSR, GDELT-GSR and AutoGSR-GSR combinations. While AutoGSR shows high correlation with the GSR time series, the same is not true for ICEWS and GDELT thereby presenting a strong case in the favor of a system such as AutoGSR to generate reliable ground truth data. Figure 7a compares the daily counts

Figure 6: Time Distribution of MITRE’s Analysts across various tasks of the GSR generation pipeline.

We requested them for an estimate of time that they spend in performing various sub tasks of the GSR generation process. Fig. 6 shows a rough sketch of the time distribution as reported by them. The figure also shows individual reductions provided for each of these sub tasks by AutoGSR. For each pie, the missing color corresponds to the provided savings: 1) Gross Discovery: 100% reduction as articles are fetched and loaded automatically into the system, 2) Skimming and Targeting: 100% reduction as articles with cumulative probability score greater than equal to 0.5 are automatically classified as protest articles, 3) Writing into GSR: almost 100% reduction as recording an extracted encoding is just a matter of few clicks, and 4) Reading and Encoding Extraction: based on the left over reduction percentage (~15%), it appears that the automatically generated encoding recommendations are providing almost 30% reduction for this task.
Table 8: Resource Distribution for GSR

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>#Resources</th>
<th>#Hours per Week</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysts</td>
<td>10-15</td>
<td>20</td>
</tr>
<tr>
<td>Q. C. Analyst</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>Manager</td>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 9: Resource Distribution for AutoGSR

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>#Resources</th>
<th>#Hours per Week</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysts</td>
<td>4</td>
<td>15-20</td>
</tr>
<tr>
<td>Q. C. Analyst</td>
<td>1</td>
<td>15-20</td>
</tr>
</tbody>
</table>

Table 10: Total Resource Hours required per Week

<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSR</td>
<td>258 hrs</td>
<td>358 hrs</td>
</tr>
<tr>
<td>AutoGSR</td>
<td>75 hrs</td>
<td>100 hrs</td>
</tr>
<tr>
<td>Reduction</td>
<td>71%</td>
<td>72%</td>
</tr>
</tbody>
</table>

Figure 7: Daily and Weekly Time Series Comparison between GSR and AutoGSR and Figure 7b compares the weekly time series for all the four systems.

6. DISCUSSION

We have presented AutoGSR, an automated event coding system for civil unrest events that is now in full continuous production use. In addition to developing and deploying the system, we have undertaken an entire life cycle analysis of how such a system would fit in an analyst’s pipeline, with a view to quantify benefits over a purely manual approach. The results from our deployment indicate that the performance measures obtained by AutoGSR are compelling to support their continued use in an event modeling setting. Future work is now focused on expanding the scope of AutoGSR to new regions of the world, and to new classes of events, beyond civil unrest.

Acknowledgments

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via DoI/NBC contract number D12PC000337, the US Government is authorized to reproduce and distribute reprints of this work for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the US Government.

7. REFERENCES


