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, 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. 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 EMEBRS 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, 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.

KDD '16, August 13-17, 2016, San Francisco, CA, USA

DOI: http://dx.doi.org/10.1145/2939672.2939737

Table 1: Sample erroneous encodings by ICEWS and GDELT.

Source	ID	Representative Paragraph	Reason for Error
GDELT	299144197	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.	Presence of the word 'demonstrate' results in a false positive.
GDELT	256666928 and 256814375	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.	No Duplicate Detection. Two events extracted from two articles covering the same story but published on different days.
ICEWS	23909784	Since its inception, the Islamic State group has demonstrated the firmness of its structure and the strength of its organizational composition.	Presence of word 'demonstrate' results in a false positive.
ICEWS	19873295	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 protest was called off.

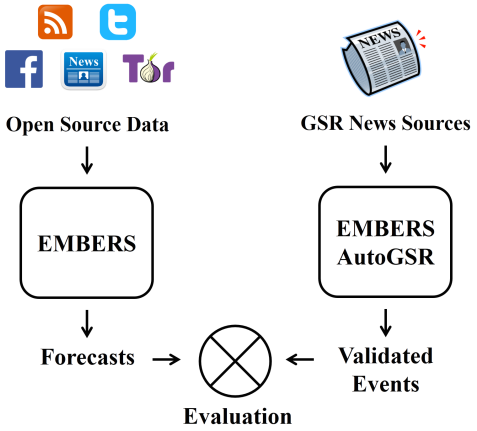


Figure 1: The EMBERS system forecasts civil unrest events from open source indicators while the EMBERS AutoGSR system encodes civil unrest events as they are reported in newspapers. Output from the AutoGSR system is used to both evaluate EMBERS forecasts and to retrain EMBERS models on a periodic basis.

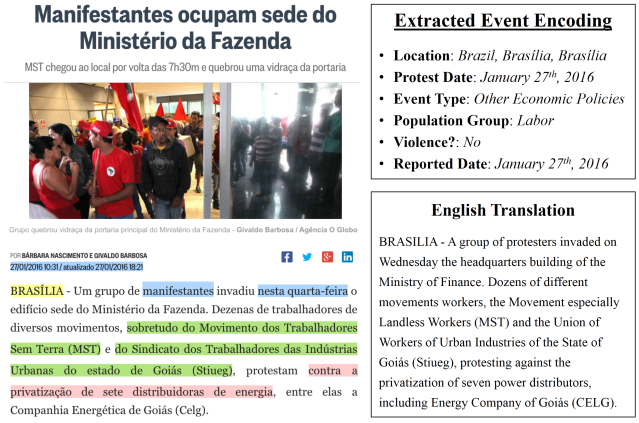


Figure 2: A sample event extracted from a news article.

Table 2: Comparison of automated event encoders.

System	Protest Encoding	Language Agnostic	Tunability	Duplicate Detection
ICEWS	✓			✓
GDELT	✓			
AutoGSR	✓	✓	✓	✓

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

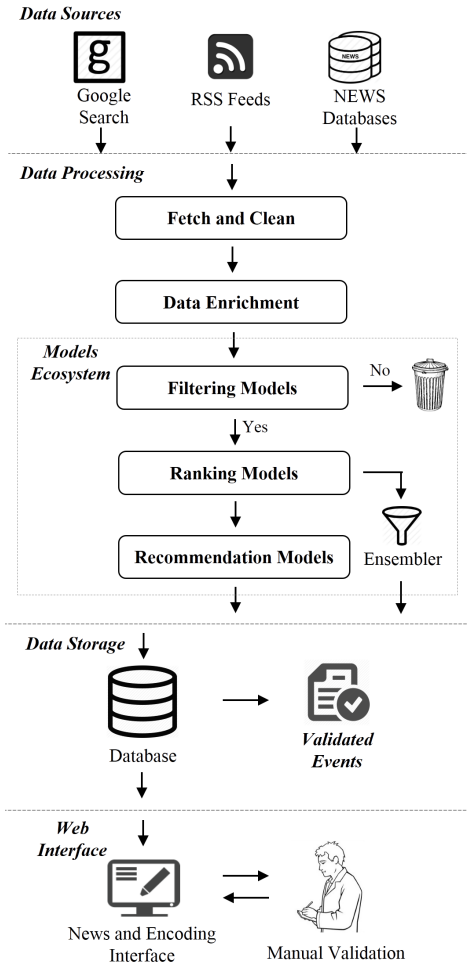


Figure 3: EMBERS AutoGSR System Architecture.

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 ensembled 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 sytem 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. Hovering the mouse over the article shows the kind of information that the system extracted from a particular sentence. 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/

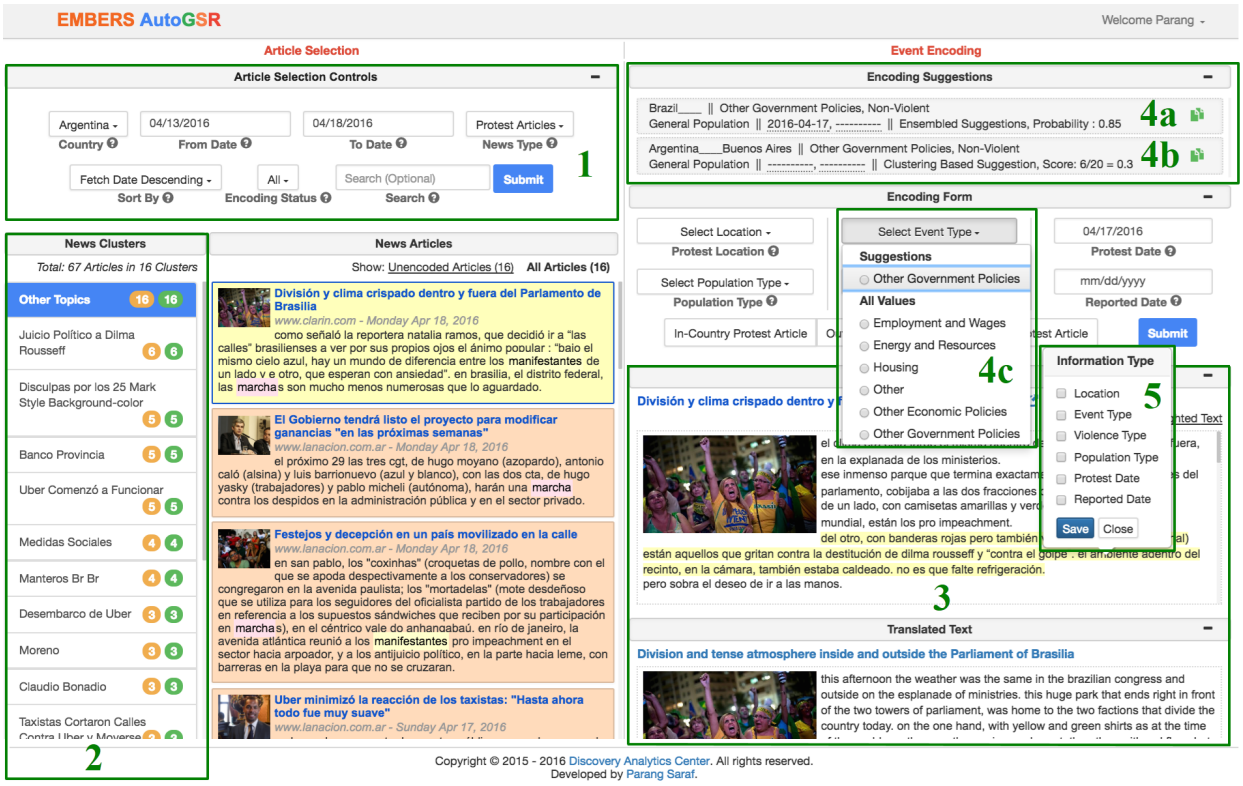


Figure 4: EMBERS AutoGSR Web Interface. 1) Filtering Controls 2) Real time Clustering of News Articles 3) Full Text, Translated Text 4) Full and Partial Encoding Recommendations 5) Model Outputs.

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 selective 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], or URLs listing stories by topic[4], or URLs corresponding to search terms[5], etc. All such URLs are considered irrelevant for the 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 assign a probability score of classi-

[3]http://www.clarin.com/politica/
[4]http://www.clarin.com/tema/manifestaciones.html
[5]http://www.clarin.com/buscador?q=protesta

Table 3: EMBERS AutoGSR: Models Ecosystem

Filtering Models	**Ranking Models**	**Recommendations Models**
These are rules based models that classify incoming news articles into protest and non protest with a 0 or 1 certainty	These models use standard machine learning algorithms on different components of an incoming article for classification.	These models assume that the incoming article is a protest article and tries to recommend complete or partial encoding(s) for the article
1. Sub Domain Based Filtering 2. URL Based Filtering 3. Negative Keywords Based Filtering	1. Labeled Unlabeled Text Classifier 2. Entity Based Classifier 3. Distributed Representation Classifier 4. MetaTags Based Classifier	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: 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 assign 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.

Table 4: Example Negative Keywords

Protest Keyword	**Negative Keyword Phrase**	**Altered Meaning**
marcha	ponar en marcha	to start; to set in motion
protesta	tomar protesta	to swear in (public official)
protesta	rendir protesta	to swear in (public official)

fying the article as a protest article. The models focus on different indicators for classifying the article.

4.2.1 Labeled Unlabeled Text Classifier

This model implements a LU Classifier as proposed by [9, 10] to work with a set of labeled and unlabeled articles. The model is adapted to work in a binary class setting and implements an Expectation Maximization procedure with a Naïve Bayes Classifier.

4.2.2 Entity Based Classifier

This model is designed to work with named entities and protest keywords and can be considered as a special case of the NB classifier built earlier. Using extracted entities and protest keywords as evidences, this model employs a multinomial naïve Bayes method with Laplace smoothing to determine the posterior probability of classifying an article as a protest article. The entities include name, location and organization along with the protest keywords found in the article.

4.2.3 Distributed Representation Classifier

This model takes a step away from the bag of words models and aims to capture the ordering of words while generating distributed vector representations for the articles. The model builds upon recent advancements in learning vector representations of words using neural networks [7, 8]. Mikolov et al. show that such vector representations not only capture syntactic and semantic information but is also aware of '*distance*' in the N dimensional representation space. The model was adapted further by [5] to generate distributed representations of documents and was shown to be *distance aware* similar to word vectors.

These models have been designed primarily to work with large datasets and have been shown in [7] to perform better with larger corpora, as they allow the layer weights to stabilize. Therefore, we train this model on an unlabeled corpus of $\sim 250k$ Spanish and $\sim 100k$ Portuguese news articles. To increase relevance, we made sure that these articles contain at least one protest keyword. To this set, we added $\sim 20k$ labeled articles. Each of the labeled articles contains two labels: 1) Unique label identifying the article; and 2) Class label identifying the article as protest or non protest. This way, we not only identify vector representations for each document but also vector representations for labels '*protest*' and '*non protest*'.

Once the distributed vector representations for '*protest*' and '*non protest*' labels have been identified, we classify an incoming article as follows: 1) Generate a distributed vector representation of the article using the trained model; 2) Calculate the *cosine distance* of the article vector from *protest* and *non protest* labels; 3) Classify the article based on closeness and assign a posterior probability.

4.2.4 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 5: Metatags used by AutoGSR.

Targeted For	Tag Name
Search Engines	title, description
Facebook	og:title, og:description
Twitter	twitter:title, twitter:description

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:

$$P(l_p) = \frac{\sum_{i=i}^{|M|} c_i o_i}{\sum_{i=i}^{|M|} c_i}$$

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 *ensembler*, that combines these individual suggestions into a complete encoding. The interface shows both complete as well as individual encoding suggestions. Lastly, the *usability models* aim to improve the ease of using the interface, by generating clustering based full encoding recommendations.

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

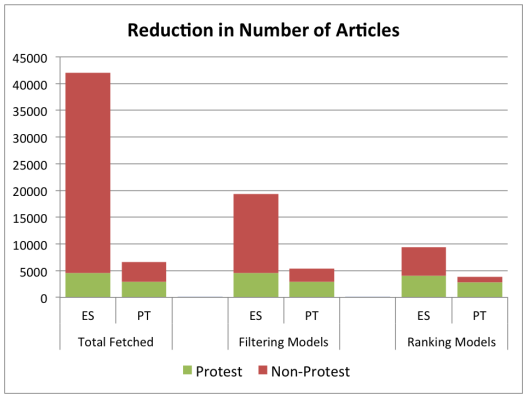


Figure 5: Reduction in number of protest and non-protest articles after each step.

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:

$$QualityScore(QS) = DS + LS + ES + PS$$

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 next:

$$DS = 1 - min(|GSR_{eventdate} - AutoGSR_{eventdate}|, 7)/7$$

In other words, if the date of the event identified by the GSR is same as AutoGSR then DS is 1. Otherwise if they are further apart by 7 days, then DS is 0.

Location score (LS) is defined as:

$$LS = 0.33 + 0.66(1 - min(dist, 300)/300)$$

where dist denotes the distance (in km) between the city identified by GSR and AutoGSR city. Both GSR and AutoGSR use standardized city names from a World Gazetteer that also provides latitude and longitude values, required for the distance computation. Cities outside 300 km get a score of 0.33; same cities get 1 and cities within 300km get scores in the range [0.33, 1].

Event type score (ES) is defined in terms of triples (e_1, e_2, e_3) where e_1 is the event granularity, e_2 is the protest reason, and e_3 is the violence status. Each of these scores have a value of 0 if they don't match and 1 if they match.

$$ES = \frac{1}{3}e_1 + \frac{1}{3}e_2 + \frac{1}{3}e_3$$

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 autogsr. 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*.

Country	Quality Score			Precision			Recall		
	Oct	Nov	Dec	Oct	Nov	Dec	Oct	Nov	Dec
Argentina	-	3.24	3.21	-	0.85	0.94	-	0.53	0.49
Brazil	-	3.53	3.58	-	0.27	0.21	-	1.00	0.99
Chile	3.24	3.19	3.35	0.53	0.90	0.87	0.84	0.81	1.00
Colombia	3.17	3.24	2.71	0.95	0.97	0.94	0.81	0.63	0.92
Ecuador	3.09	3.20	3.20	0.60	1.00	0.86	1.00	0.84	1.00
El Salvador	3.33	3.09	3.10	1.00	0.89	0.80	0.91	1.00	1.00
Mexico	3.13	3.36	3.16	0.92	0.73	0.93	0.95	0.83	0.86
Paraguay	3.60	3.42	3.39	0.74	0.82	0.78	1.00	1.00	1.00
Uruguay	3.17	3.07	3.12	0.53	0.77	1.00	1.00	0.85	0.73
Venezuela	-	3.48	3.39	-	1.00	0.86	-	0.59	0.93

*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.

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

2015, Nov 2015, and Dec 2015. Here recall corresponds to the number of events identified by the GSR that were also identified by AutoGSR; and precision corresponds to the number of events identified by AutoGSR that were also found in the GSR. As the results in Table 6 demonstrate, the AutoGSR system has a near perfect precision and recall, thereby indicating that the system identified almost all civil unrest events reported by the GSR while maintaining a really low false positive rate. Further, for these identified events, the quality of the extracted encodings is very close to the true GSR encodings.

Brazil is the only country which appears to have a high false positive rate which can be attributed to an incorrectly added news source. *Globo* is a big media conglomerate in Brazil that owns newspapers, news channels, and online real time news portals. While MITRE focuses only on the electronic version of the print newspaper (*oglobo.globo.com*), the AutoGSR system was fetching articles from **.globo.com*. These also included articles from the real time news portal *g1.globo.com* which reports events in real time without much verification. Hence, we identified several more protests because the evolving news stories were considered as different events. This is also corroborated by the fact that many of the g1.globo.com links are dead or have a short life span as links pointing to intermittent stories are removed once all the details about an event are known. Interestingly, Brazil has the highest Quality Score with a perfect recall pointing to the fact that all the events reported by GSR were still correctly identified by AutoGSR, and with very high quality. These extraneous sub domains were removed from the system in mid January 2016.

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.

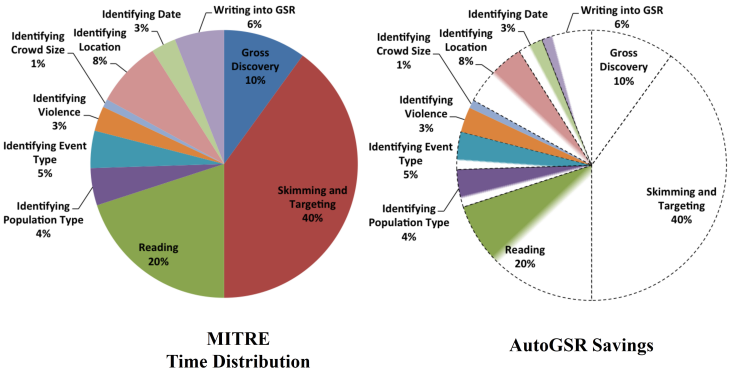


Figure 6: Time Distribution of MITRE's Analysts across various tasks of the GSR generation pipeline.

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 33% reduction for this task.

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

Table 8: Resource Distribution for GSR

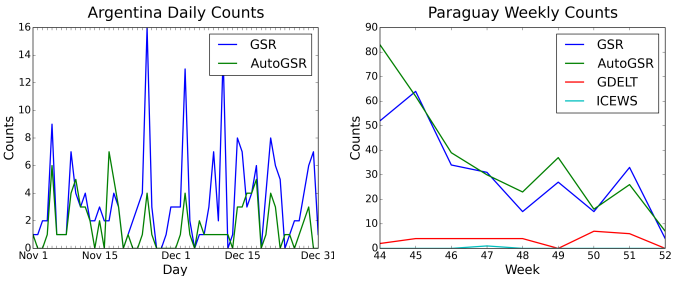
Resource Type	#Resources	#Hours per Week
Analysts	10-15	20
Q. C. Analyst	2	24
Manager	1	10

Table 9: Resource Distribution for AutoGSR

Resource Type	#Resources	#Hours per Week
Analysts	4	15-20
Q. C. Analyst	1	15-20

Table 10: Total Resource Hours required per Week

	Min.	Max.
GSR	258 hrs	358 hrs
AutoGSR	75 hrs	100 hrs
Reduction	71%	72%



(a) Argentina Daily Counts Comparison

(b) Weekly Time Series Comparison for Paraguay

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

[1] E. Boschee, P. Natarajan, and R. Weischedel. Automatic extraction of events from open source text for predictive forecasting. In *Handbook of Computational Approaches to Counterterrorism*, pages 51–67. Springer, 2013.

[2] C. Cortes and V. Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, Sept. 1995.

[3] F. Hogenboom, F. Frasincar, U. Kaymak, and F. De Jong. An overview of event extraction from text. In *DeRiVE Workshop at ISWC 2011*, volume 779, pages 48–57. Citeseer, 2011.

[4] T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *ECML '98*, ECML '98, pages 137–142, London, UK, UK, 1998. Springer-Verlag.

[5] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

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

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

[8] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[9] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Learning to classify text from labeled and unlabeled documents. In *AAAI '98*, AAAI '98/IAAI '98, pages 792–799, Menlo Park, CA, USA, 1998. American Association for Artificial Intelligence.

[10] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Mach. Learn.*, 39(2-3):103–134, May 2000.

[11] 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.

[12] S. Osinski and D. Weiss. A concept-driven algorithm for clustering search results. *IEEE Intelligent Systems*, 20(3):48–54, May 2005.

[13] N. Ramakrishnan and P. Butler et. al. 'beating the news' with embers: Forecasting civil unrest using open source indicators. In *KDD '14*, KDD '14, pages 1799–1808, New York, NY, USA, 2014. ACM.

[14] L. Ramshaw, E. Boschee, M. Freedman, J. MacBride, R. Weischedel, and A. Zamanian. Serif language processing effective trainable language understanding. *Handbook of Natural Language Processing and Machine Translation: DARPA Global Autonomous Language Exploitation*, pages 636–644, 2011.

[15] P. A. Schrodt. Tabari: Textual analysis by augmented replacement instructions. *Dept. of Political Science, University of Kansas, Blake Hall, Version 0.7. 3B3*, pages 1–137, 2009.

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