Information Extraction and Summarization System

Aniket Kulkarni[1]
Computer Science
University at Buffalo
Buffalo, New York, USA
aniketvi@buffalo.edu

Pranav Bhagwat[2]
Computer Science
University at Buffalo
Buffalo, New York, USA
pbhagwat@buffalo.edu

Soumitra Alate[3]
Computer Science
University at Buffalo
Buffalo, New York, USA
smalate@buffalo.edu

ABSTRACT

With the growth of digital media, articles and news who has the time to go through the entire articles and news? This volume of text is an invaluable source of information and knowledge which needs to be effectively represented and summarized. Therefore, we propose to design and implement an efficient as well as an easy to use event extraction and summarization mechanism. Our system design ensures to provide detailed information about the events related to political violence and riots, making it possible for the user to gain meaningful insights regarding the event just by skimming through the summary by ingesting news articles.

System generates following information for each article: event date, event location, event type, parties involved, data sources and a brief description. The event types which we have focused on are Riots, Protests and Violence against civilians. A list of countries for which we are performing the task of violence activity detection and summarization are India, Indonesia, Thailand. Our evaluation demonstrates the effectiveness of the system.

CCS CONCEPTS

Information extraction, summarization

KEYWORDS

Event Extraction, Classification, Summarization, Evaluation

1 INTRODUCTION

With the increasing amount of unstructured textual data and exploding number of digital news data, extracting information and decision making process becomes difficult motivating the need for system that can extract and categorize important event. We describe a system for scraping news articles and extracting events related to political violence such as protests, riots and violence against civilians from India, Indonesia and Thailand new articles. We outline a comprehensive architecture which will identify, categorize and summarize the data. The system will categorize the unstructured data based on event type, event location, event date, parties involved in the event, data sources and a brief summary of the articles. The task Automatic summary generation is task of preserving key content and original meaning.

2. SYSTEM OVERVIEW

The high level representation of the system is -

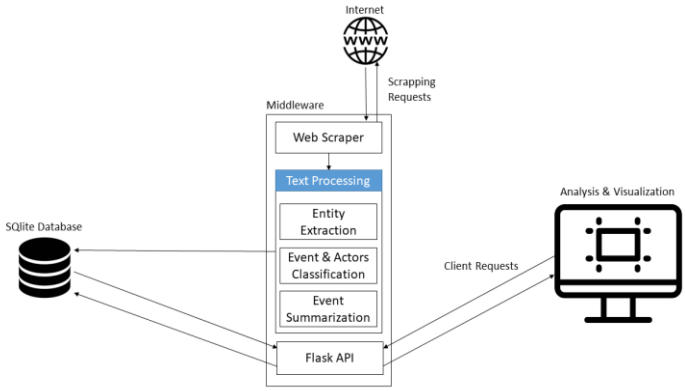


Figure 1: System Architecture Overview

2.1 Scraping the data from web

We are making use of python Newspaper3k library to scrape the data from web. The major sources which we have used to scrape the data are as follows-

Sources for India: Hindustan Times, India Today, Business Standard, FPJ, Times of India

Sources for Thailand: The Nation, Khaosod English, Intellasia, Thaiger, Bangkok Post, Al Jazeera, The Star,The Guardian, Asean Today, The Diplomat, Washington Post

Sources for Indonesia: Jakarta Post, Strait Times, Al Jazeera, Reuters,Tempo, Vatican, Nikkei Asian Review, Brookings, The Economist

We are storing the links and RSS feeds related to the above sources and scraping the articles on these links.

2.2 Entity Extraction

After scraping the data, our next step is to parse every article and extract relevant fields like title, description, location, date, event type and parties involved. The system description section of the report provides some details about different

packages that we have used for extracting the above individual entities.

2.3 Events and Actor Classification

At this step, we are performing classification of events and parties involved in an article using supervised learning models such as Bag of Words, Naive Bayes, SVM and Random Forest.

2.4 Event Summarization

After extracting the required entities from events and classifying the events, we summarize the description of the filtered articles using LexRank algorithm. Further details regarding this is explained in the next section.

2.5 Analysis and Visualization

The filtered and summarized events with their entities are stored in SQLite database. We make use of python Flask as middleware for pulling this data from database and passing it to front end.

3. SYSTEM DESCRIPTION

A detailed description of system architecture is presented in figure 2. The input rss news feed and unstructured news articles are scrapped using python newspaper3k library. Then the scraped data is given to named entity tagger spacy and GeoText for extracting various entities such as Location, Parties Involved. For classifying the event type we are using random forest classifier which is trained on ACLED dataset.

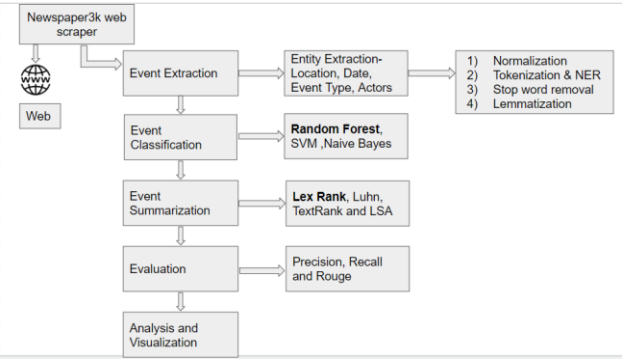


Figure 2: Detailed System Architecture

3.1 Event Extraction:

A recurring subject in NLP is to understand large corpus of texts through topics extraction. Whether you analyze the news, digital data, understanding key topics will always come in handy. Our system makes use of python library spaCy for performing event extraction. SpaCy assign labels to group of tokens. It provides a default mode which can recognize a wide range of entities which include person, organization, language, event, etc.

Another named entity tagger which we tried in our system is Stanford NER which is a named-entity recognizer based on linear chain Conditional Random Field (CRF) sequence models.

Both spaCy and Stanford NER models can be used for named entity recognition on unstructured documents achieving reasonably good outcomes. The former has the advantage of automatically recognizing the entities out of the persons' tokens.

3.2 Text Summarization:

The most important sentence from news articles are represented as summary. The generated summary is extractive which means the system rely on extracting several parts, such as phrases and sentences, from a piece of text and stack them together to create a summary. Our system makes use of the algorithm named LexRank based on the same ideas behind Google's PageRank algorithm. LexRank is an unsupervised approach that uses IDF-modified Cosine as the similarity measure between two sentences. This similarity is used as weight of the graph edge between two sentences. The post processing step in LexRank makes sure that that top sentences chosen for the summary are not too similar to each other. Other popular text summarization includes Luhn, LSA, TextRank.

Luhn algorithm makes use of naive approach based on TF-IDF and looking at the window size of non-important words between words of high importance. It also assigns higher weights to sentences occurring near the beginning of a document

Latent semantic algorithm uses an unsupervised method of summarization; it combines term frequency techniques with singular value decomposition to summarize texts.

Text rank is a graph-based summarization technique with keyword extractions in from document.

3.3 Classification of events and parties involved in an event

We propose to employ a supervised learning model to generate classification predictions. We created training and testing data sets for the classification of events and parties involved from the data made available by ACLED. For both classification tasks, we followed the following steps:

(1) Transform the datasets available for each country made available by ACLED to reflect the summary, either event type or parties involved based on the type of classification you wish to perform and shuffle the entire dataset.

(2) Normalize, tokenize and lemmatize the summaries in the dataset and generate vector representations for each summary, one hot encode the target labels and split the data into training and testing data sets.

(3) Fit the training dataset to the model and make predictions on the testing dataset and evaluate accuracy.

3.4 Identification of an event-specific location

Our goal for this task is to extract the most relevant location with respect to the free text under consideration for a given article. An article may contain multiple locations, some of which may be relevant to the event while the rest could be considered as of relatively less relevance; our goal is to extract specific locations. We proposed to identify labels described as either 'GPE' or 'LOC' in the tokenized text. We have developed a solution similar to ensemble methods by employing spaCy together with GeoText to identify and tag the location that is of most relevance to a given article. We are assuming that the location which the article mentions the most number of times, is most likely the relevant location about the given event in the article. Tagging the correct event is a difficult task due the ambiguity of location names.

4. EVALUATION

This section focuses on the results of the experimental studies that we have conducted on our news scraping and information extraction system. We are reporting statistics for a set of 100 random instances from the data generated by ACLED against our proposed system. We have evaluated the results for the event type, parties involved, and location with the help of precision, recall and f-measure. We have evaluated the extractive summary with the help of the Rouge metric which is also defined in terms of precision, recall and f-measure.

4.1 Event Type

In this section we evaluated the performance of our proposed system in the task of classifying an event as either Riots, Protests or Violence Against Civilians against the event type classified by ACLED for the exact same news. We developed a baseline solution around the bag of words concept. We manually created a bag of words which contained the tokens that described the event types we are focusing on viz., riots, protests and violence against civilians. If the tokenized text contains any token present in the event type bag of words, we will classify the event respectively. Further, we trained three different supervised classification models viz., Multinomial Naive Bayes, SVM and Random Forest. Through our experiments it was observed that while the bag of words is perhaps the most basic concept in information extraction it generated decent performance over the test set for our baseline model. However, the supervised classification models outperformed the bag of words model with ease. Over the testing set, the results of our solution indicated that the Random Forest classifier recorded the highest scores for our evaluation metric.

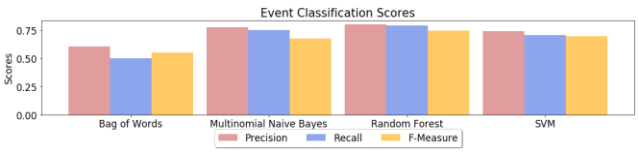


Figure 3: Statistics about classification of events on test data set

4.2 Parties Involved

In this section we evaluated the performance of our proposed system in the task of identifying and tagging the type of parties that were involved in a given event which is described by the article under consideration. Similar to classifying an event type for a given article, our baseline system was developed around the bag of words concept. We manually created a bag of words which contained the tokens that described the most influential political parties for a given country under consideration and identified that a party is involved if the tokenized text contains a key word contained the bag of words. We trained three different supervised classification models viz., Multinomial Naive Bayes, SVM and Random Forest. The results of our experiments showed that while the bag of words baseline model generated better precision, the parties classified by the models generated higher recall and f-measure but at the cost of lower precision than the bag of words model.

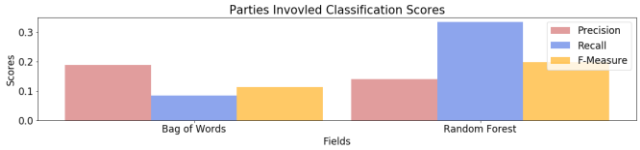


Figure 4: Statistics about classification of parties involved on test data set

4.3 Location

In this section we evaluated the performance of our proposed system in the task of identifying the location at which the event under consideration has taken place. Initially the proposed system only used spaCy. The baseline system performed entity tagging over the entire article text and extracted entities which were tagged as 'LOC'. However, we realized that spaCy failed to classify certain location entities as 'LOC". We shifted our focus to the Stanford NER as an alternative to spaCy and also developed a solution which uses spaCy and the Stanford NER together. Through our experiments we observed that using spaCy and the Stanford NER gave us better results. We considered the system that used spaCy together with Stanford NER as the baseline for our location identification task. To improve upon the baseline system, we developed a solution that uses spaCy and GeoText, a python package, together. We evaluated the new proposed solution over the same test data set and observed an increase in all three metrics of evaluation, precision, recall and f-measure.

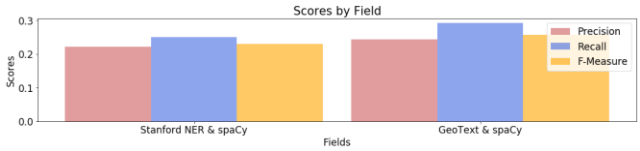


Figure 5: Statistics about identification and tagging locations on test data set

4.4 Summary

In this section we discuss the evaluation of the performance of our proposed system in the task of generating an intuitive summary from the free text contained in the news article. Our first focus was to generate an extractive summary based upon the importance of key words in a sentences in the text under consideration. We developed two solutions based upon the frequency of terms, Luhn summary generation algorithm and LSA. To improve upon the baseline system, we implemented centroid and centrality based approaches, Text Rank and Lex Rank. Our experiments revealed that Lex Rank had far superior performance than all the other summary generation methods. We observed an approximate 9% increase between our baseline and current summary generation solution in the rouge evaluation metric.

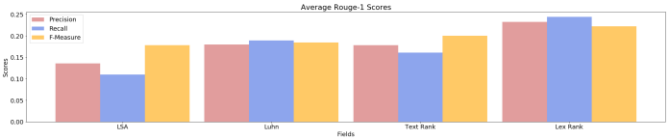


Figure 6: Statistics about the average Rouge-1 scores for extractive summaries generated by the system

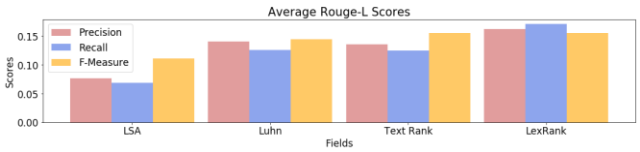


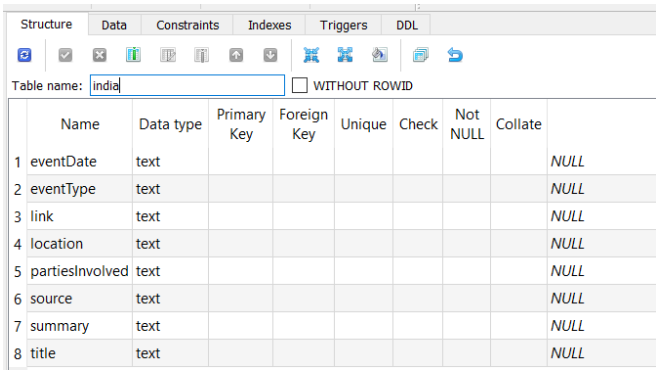
Figure 7: Statistics about the average Rouge-L scores for extractive summaries generated by the system

5. Analysis and Visualization

This section of the report talks about the system analysis and visualization part of our system.

5.1 SQLite Database

SQLite is an in-process library that implements a self-contained, server-less, zero-configuration, transactional SQL database engine. We scrape the web, filter the data using the different classification algorithms to retrieve the articles related to political violence, craft a concise summary of the article and finally store all these details in database tables. We have created a database and multiple tables in the database to store the country-wise article details separately. We have created 4 tables in total. There are three tables viz. 'india', 'thailand', 'indonesia' to store the country-wise information. Table 'agg' stores combined data of all the countries and it is used for corpus analysis. We query on these tables to perform the visualization.



	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
1	eventDate	text							NULL
2	eventType	text							NULL
3	link	text							NULL
4	location	text							NULL
5	partiesInvolved	text							NULL
6	source	text							NULL
7	summary	text							NULL
8	title	text							NULL

Figure 8: Sample table overview using SQLite Studio

5.2 Flask Framework and Jinja2

Flask is a web application framework in python and Jinja2 is a popular template engine integrated with Flask. It combines a template with a certain data source to render dynamic web pages. The flask acts as a middleware in our system. It is responsible for making a connection with database, retrieve data from database, converting the data in JSON format and passing the data to front end where we are finally displaying the visualization.

5.3 Frontend Technologies

We are making use of HTML5, CSS3, JavaScript and Bootstrap framework to build web application with scalable and responsive user interface.

5.4 Analysis

5.4.1 India

5.4.1.1 Classified event representation

We are displaying the events related to political violence for India as seen in the figure below. We have given the functionality to display this classified data by applying further filters. This functionality includes filtering of data scraped in last 24 hours, last 2 days and Last week. If we do not apply any filter, all the historical data collected by the system till current date is displayed. We are extracting and displaying the Event Title, Event Summary, Event Type, Event Location, Event Date and the Parties Involved in the activity. We have also calculated the count of total political violence events per Event Type which is displayed in the tile below the filter section.

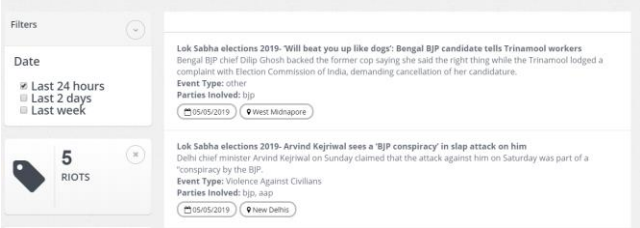


Figure 9: Classified event representation and a brief summary

5.4.1.2 Distribution by top political violence prone locations:

We are displaying the map of India and calculating the top locations where the proportion of political violence events is maximum which gives a good insight about which locations to look out for when analyzing the election violence.

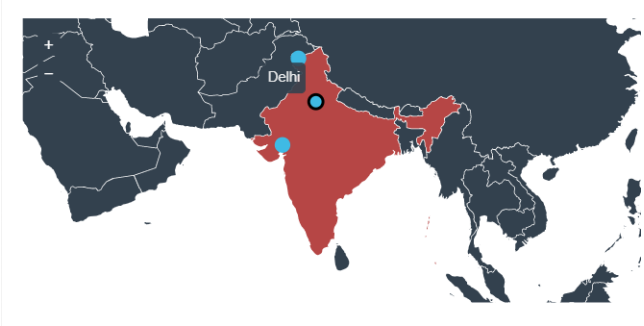


Figure 10: Distribution by top political violence prone locations

5.4.1.3 Bar chart for distribution by city

Here we are visualizing the cities/locations where the Indian political violence activities take place very often in the form of bar chart. By observing the chart below, as Pakistan is also seen as a top location, we also infer that the Indian news articles are reporting cross-border events arising due to political activities taking place in India. The top 5 locations for India are India, Delhi, Pakistan, Ahmedabad and Kashmir.

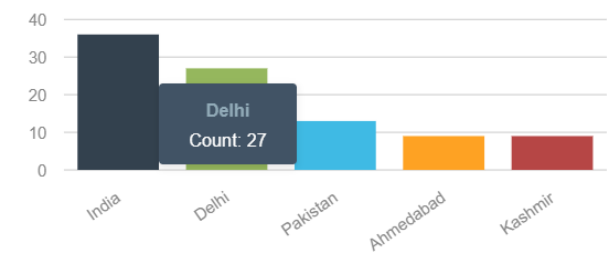


Figure 11: Bar chart for distribution by city

5.4.1.4 Distribution by top 5 Parties Involved

Here we have analyzed and extracted the top 5 parties which have major contribution in political violence related activities. Such top 5 parties for India are bjp, congress, combination of bjp & congress, protesters and rioters.

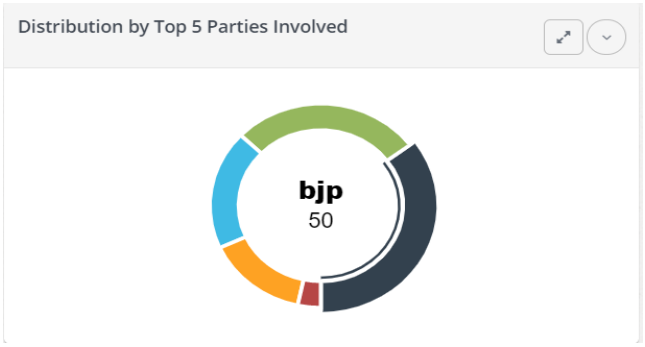


Figure 12: Distribution by top 5 Parties Involved

5.4.2 Thailand

5.4.2.1 Classified event representation

We are displaying the events related to political violence for Thailand as seen in the figure below. We have given filtering option similar to India for this page. We inferred that it is difficult to find sources to retrieve relevant data for Thailand and we had to overcome multiple challenges while scraping data for this country.

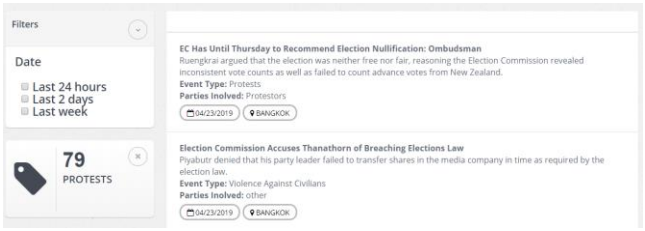


Figure 13: Classified event representation and a brief summary

5.4.2.2 Distribution by top political violence prone locations

We are displaying the map of Thailand and calculating the top locations where the proportion of political violence events is maximum which gives a good insight about which parties and locations to look out for when analyzing the election violence.

Figure 14: Distribution by top political violence prone locations

5.4.2.3 Bar chart for distribution by city

Here we are using bar chart representation to display the top locations with maximum political violence activity. The top 5 locations for India are Bangkok, Thailand, Chiang Mai,Huai Khwang and New Zealand. We can infer that New Zealand is also one of the participants contributing to political violence.

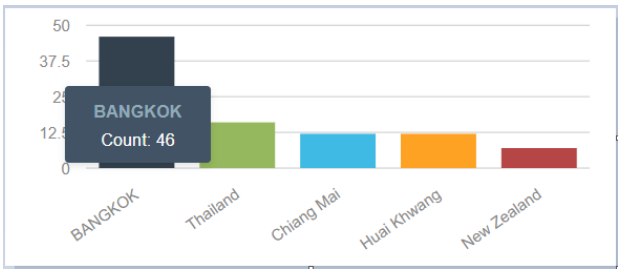


Figure 15: Bar chart for distribution by city

5.4.2.4 Distribution by top 5 Parties Involved

Here we have analyzed and extracted the top 5 parties for Thailand which have major contribution in political violence related activities. Such top 5 parties are Future Forward, Pheu Thai, Protesters, Democrat and Bhumjaithai.

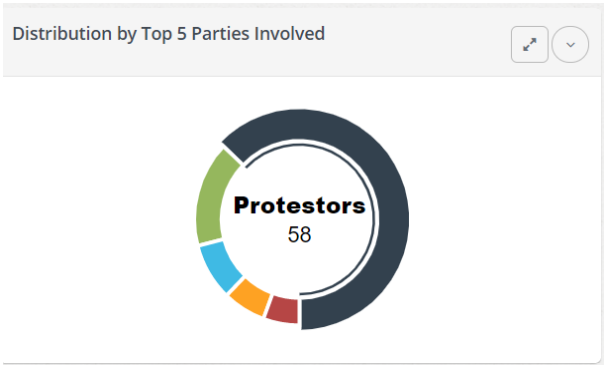


Figure 16: Distribution by top 5 Parties Involved

5.4.3 Indonesia

5.4.3.1 Political violence event summarization

We are displaying the events related to political violence for Indonesia as seen in the figure below. We have given filtering option similar to India and Thailand for this page as well. Scraping the data for Indonesia posed some additional challenges for us as Newspaper3k library has very limited support for extracting the Indonesia related data.

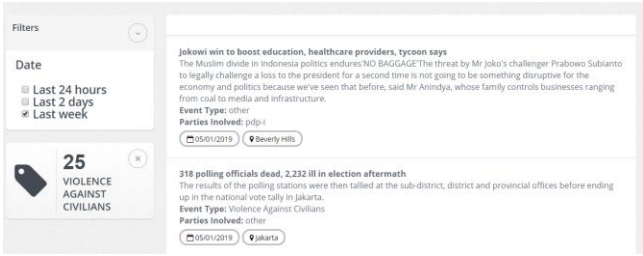


Figure 17: Classified event representation and a brief summary

5.4.3.2 Distribution by top political violence prone locations

We are displaying the map of Indonesia and calculating the top locations where the proportion of political violence events is maximum. As shown in the map below, we observe that the map of Indonesia is scattered but the political violence activities are concentrated within limited areas in eastern region.

Figure 18: Distribution by top political violence prone locations

5.4.3.3 Bar chart for distribution by city

Here we are using bar chart for visualizing the cities/locations where the Indonesian political violence activities take place very often. By observing the chart below, we infer that the Indonesian election violence activities take place outside Indonesia as well. The top 5 locations for India are Jakarta, Indonesia, Bogor, Aceh and Malaysia.

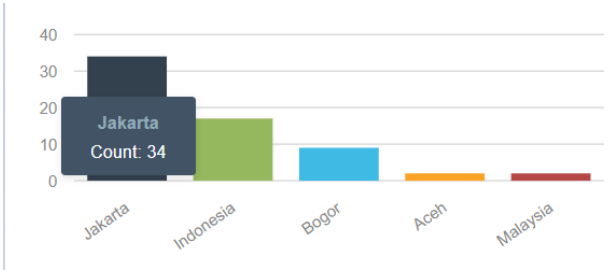


Figure 19: Bar chart for distribution by city

5.4.3.4 Distribution by top 5 Parties Involved

Here we have analyzed and extracted the top 5 parties which have major contribution in political violence related activities in Indonesia. Such top 5 parties are PDP-I, Gerindra, combination of PDP-I & Gerindra, Rioters and Protestors.

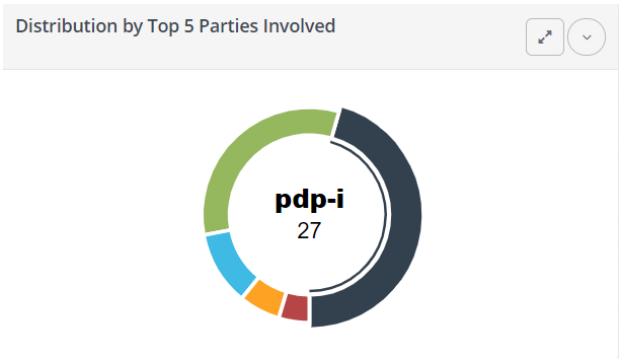


Figure 20: Distribution by top 5 Parties Involved

5.4.4 Aggregate Cross Country Analysis

5.4.4.1 Distribution by county

We have calculated and displayed the total number of political violent activities that have taken place in the counties of India, Thailand and Indonesia. For example, total 247 political violent activities have occurred in India.

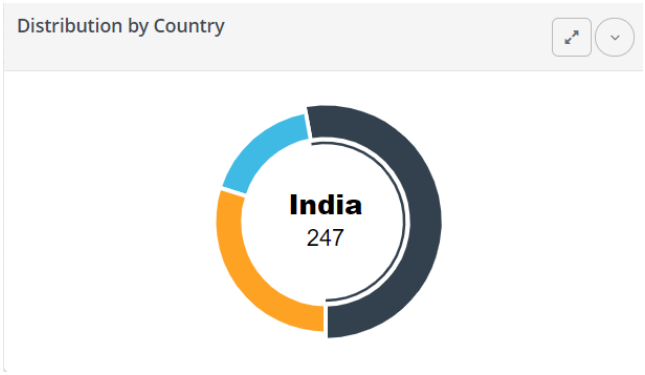


Figure 21: Distribution by county

5.4.4.2 Distribution by Event Types

Here we are displaying the total number of violent activities per event type across all the three countries. For Example, the violent political activities with type categorized as ‘Violence Against Civilians’ has a total count of 214 across all the three countries. It is very important indicator of what kind of activities have been recorded in majority. Predicting this type, it is possible to employ future actions to be taken to avoid such activities.

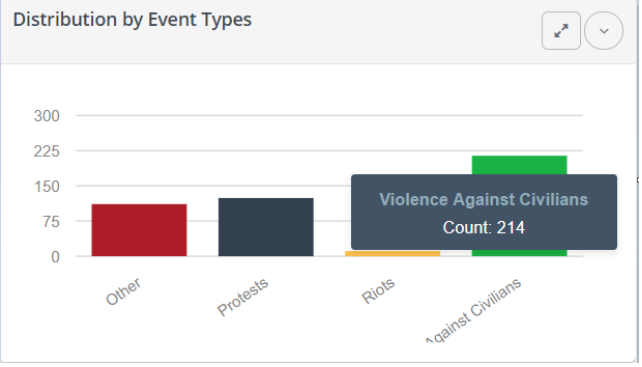


Figure 22: Distribution by Event Types

5.4.4.3 Events Timeline

In order to provide the user, some important insight about comparison of each type of violent event taking place on each day and to give a historical representation, we are making use of the timeline visualization. For example: as seen in the figure below, we can infer that on 1st May 2019, 11 protest activities,1 Riot and 30 Civilian violence activities took place. This information is important perform trend analysis by observing the changes in the count and type of violent events over the specified date range.

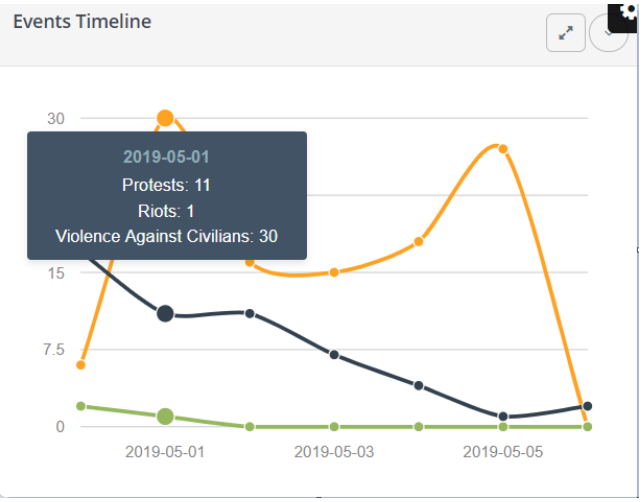


Figure 23: Time Series Distribution by Event Types per week

5.4.4.4 Sliding tiles for country wise violence count

Here we are calculating the total number of violent events belonging to each Event Type for each country. As seen in the figure below,145 events with event type as Violence Against Civilians have been reported. Similarly, 25 related events in Indonesia and 44 events in Thailand.

Figure 24: Sliding tiles for Country wise violence count

CONCLUSION

We have successfully designed and implemented an end-to-end event extraction and summarization system. We have focused on extracting, classifying events by training our system using classification algorithms and summarizing political violence related activities for the countries India, Indonesia and Thailand from news articles and make it available to a client interface.

REFERENCES

[1] https://towardsdatascience.com/text-summarization-in-python-76c0a41f0dc4
[2] https://www.tutorialspoint.com/sqlite/sqlite_overview.htm
[3] https://www.tutorialspoint.com/flask/flask_overview.htm
[4] https://rare-technologies.com/text-summarization-in-python-extractive-vs-abstractive-techniques-revisited/
[5] https://nlp.stanford.edu/ner/
[6] https://spacy.io/
[7] https://www.sqlite.org/index.html
[8] http://flask.pocoo.org/
[9] https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd
[10] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html