Planned Event Forecasting from News Feed

Neelesh K Shukla (164101009)
Neha Saini (164101025)
Pranay Sanghvi (164101052)
Suweta Shakya (164101061)

1. ABSTRACT

People *s*tudying events are mostly focused on event detection where they work on identifying event which has been already occurred (retrospective) or happening right now (Online). We wanted to take a step further and work on identifying events that are likely to occur in future. (Event Forecasting).
In this project we built a system to identify planned events of protest and forecast them. We have collected the articles from news feeds and filtered the events of protest.
Once the article is identified for ongoing or planned protest, some useful information like Title, Date, Location, Person and Organization involved, have been extracted.
Next we normalized the date expressions to locate the planned protest related events and displayed using web based user interface.

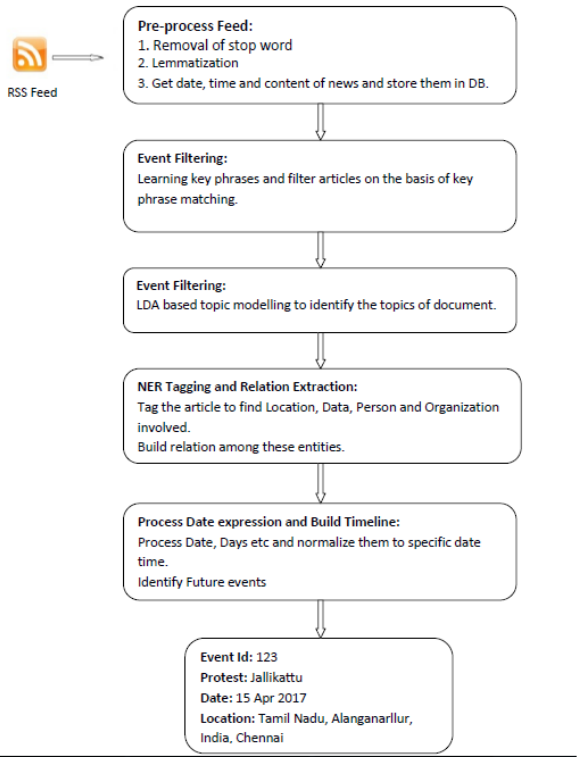
2. INTRODUCTION

India is the largest democracy in the world and with so much diversity comes so many demands. Civil unrest is a common occurrence in our country. We want to develop a system which can predict any such happening in the future. This will help official to take necessary action.

There can be two flavours of event forecasting: Unplanned and Planned. For now, we are focusing on Planned Event Forecasting and that events will be of type protest, marches etc.
To build protest forecasting model, our first aim is to find all the protest related data. There are enormous sources for such data like news websites, social networking sites etc. One challenge with social networking sites is variation of data, different person uses their language which may or may not be in structured form. Since news website data is in structured form so here in our models we will be working with some well known news websites for extracting and filtering relevant data. Once we have filtered data next step is to use phase filtering to find a set of key phrases which will identify give us protest related information. Based on these key phrases we will parse the relevant articles and tag them with location, date, time, organization etc.
The process for extraction and tagging of data will work on number of different articles, so the whole corpus will contain various location, organization name, date etc. This will lead us to our next step of identifying data of interest. For eg. Out of multiple locations that we will get from tagger, distinguish actual location of protest and similarly finding correct protest date, protest type etc. Once we have the actual date of protest we can easily work with it to find whether it is a future event or not.

3. METHOD

In general, our approach is to get the html pages from news feed, pre-process them to get the article publish date, title and content, identify the articles indicating the event of social unrest especially protest, extract relevant information to forecast events.
High level diagram is shown as below:



3.1 Building Corpus

First step to build the data set was to parse the news feeds of multiple news websites, as we needed information about planned protest we found news websites as most reliable source rather using Facebook, Twitter or any other social network sites So to build the data set we need to chose the

websites to get the articles. We Chose 4 news Websites viz Hindustan Times, Times of India, The Hindu and Indian Express as they were publishing significant number of articles per day.

3.1.1 Fetching the news articles:

We have parsed the news feeds from 16th January 2017 to 25th of march. Along with the content of article we also stored Date of publishing the article and Title of the Article. To serve this purpose we used Feed Parser which returns the list of handles of all the news articles present in the feed, and then we iterated over that list to fetch all the web pages, and stored them as ".HTML" file.
Out of all the processed articles, we could find only few articles on protest. We needed more articles which contains protest related data to work with our models. So we took data from OSINT lab.

3.1.2 Extracting attributes of articles:

Next step was to extract the information like Date of publishing the article, Title of the article and main content of the article etc. To serve this purpose we used attributes of feed Parser Dictionary such as date , time , summary , title etc and then stored them into the Database using SQL.

3.1.3 Linguistic Pre-processing

Once the articles are extracted from the news article we have done few pre-processing like removal of punctuation marks, removal of stop words and lemmatization.

3.2 Relevant Document Identification

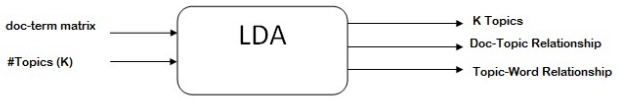
In general there are multiple words or phrases which can be used to address same event because of having similar context i.e. instead of using word **protest** different article uses word having same meaning as protest. In key phrasing we need to list out all such words. To do this task initially we have found a set of words with same context as protest based on our domain knowledge which are PROTEST, DEMONSTRATION. Similar method has been used by a team of Virginia Tech.[3]
There can be many other words which may be used in the same context. There are two ways to learn all such words. One way is to find all the synonyms of words protest and demonstration. This can be done by using WordNet[8]. This approach has two problems. One the synonyms may not be relevant like presentation, test, seminar. Other issue was that synomys miss the domain specific or the terms used in local languages like Dharna, Bandh etc. Another way is to find all the words which shares same context as protest is to learn these key phrases. So we built a key phrase learning model as described below.
We have followed two approaches to do this task. In our first approach we used Latent Semantic Analysis (LSA) technique to learn such word. We applied SVD on the word-word concurrence matrix to learn word embeddings with window of 10 words. [6]
We applied cosine similarity on resulting words vectors to get the words similar to identified 3 words i.e. PROTEST, STRIKE, DEMONSTRATION. Due to a large amount of variation in data we could not find good results with this approach so we moved to another approach which is word2vec.

3.2.1 Word2Vec

In second approach we have used word2vec model. We used our collected news articles as corpus for applying it to generate word embeddings. To get the words with same context we took words 'protest' and 'demonstration' as target words. We then identified the relevant documents which contained these key phrases.
In next step we wanted more refined articles.
Reason of extracting more refined results: Out of all collected document from key phrased learning we found that there were some document which had above filtered words but the documents were not relevant to us. This is so because of having ambiguity in meaning of words, for eg- word strike can be used either in context of war or in protest. So to remove these kind ambiguity and get more refined result we have implemented LDA model.

3.2.2 LDA Model

LDA [1] model is used identify hidden themes (topic) in document. Each document can be thought of as mixture of topics and each topic can be considered as mixture of words.



We used vector space model representation of doc, which is very frequently used in solving many kind of Information retrieval problems. In Vector space model a document can be represented as n-dimensional vector where n is vocabulary size. d(w1, w2,...., wn) Each entry d_{ij} is weighed term frequency of a word w_j in document d_i. We have followed below procedure to implement LDA.
Initially for all the key phrase filtered document we applied lemmatization, removed stop words and also removed all those words which had 1 occurrence in all documents since clearly these words would not be having much contribution in document. After finding highly occurrence words we built term document matrix and then calculated TF-IDF matrix to weight words according to their importance in document. With TF-IDF matrix we built topic model and parse them to find protest relevant topics. For these filtered topic we filtered documents which had high probability of chosen topic.

More details on implementation are explained in Experiment and results section.

3.3 Information Extraction

As next step we want to extract relevant information from articles. Our goal to get the location, data and person/organization involved as below.
[Date, Location, Person, Organization]

To extract above information for all the documents we had to do tagging which will identify person 's name, location and organization in all the documents.

We used sequence based model for labeling the words with appropriate labels. We used Stanford 7 class NER tagger[2] to tag PERSON, LOCATION, DATE, ORGANIZATION. This implementation uses CRF based sequence model.
An article contains many entities which may not be useful as they may be peripheral information not actually involved in protest. For eg: A line like 'Narendra Modi<PERSON> called a meeting to discuss issue raised in protest' has person tag, but Narendra Modi is not involved in protest to not relevant.

As next step we need to form relations between entities to identify related entities. We build following relation triplet which have some special relations.
<First Part containing Entity1, Verb, Second Part containing Entity2>

We used Stanford Relation Extractor to build these kind of triplets which implements algorithm described in paper [9]. We got relations like:

(jpp<organization>, call on, february<date> 16<date>)
(jpp<organization>, call for, Jharkhand<location> bandh)

Once these relations are build, relevant information needs to be extracted. For this purpose we designed an algorithm described as below.

Pattern Matching:
This algorithm looks for pattern present in relation and process it for extracting information.
Pattern finding will not work for above use case of JPP calling bandh. Assume we have a pattern like '<ENTITY> calls for bandh' will skip the first relation. But this relation has date in it which is relevant information.

Window Based Information Extraction: We overcome this issue by using concept of spatial locality. Usually the nearby sentences are very closely related in any news article. The relation extractor using sequence based model, if a relation has some important information the nearby relation also will have relevant information. We used window size of 3 relations to extract information.
Algorithm:

1. Build pattern for verb phrase and look for them in relation.
2. Create a separate list of each type of entity
3. Start processing relations sequentially
4. Pick the relation and relations in its window. Locate entities, if no entity present in the relation, mark that relation processed and go for next relation.
5. If relevant entity present extract person, date, location and organization.
6. Check if the entity is already present in its respective list and if so, ignore entity, otherwise add them in their respective list and mark relation processed.
7. Continue above steps till there is any unprocessed relation.
8. Mostly titles of news articles have most relevant information, check tagged title of article for the patters defined in first step and extract information.

We ignored co-reference and entity disambiguation issues.

4. EXPERIMENT

4.1 Dataset

As described in earlier sections we are downloading news article from different news websites using feeds and storing them on our local drive.
We crawled the data from Times Of India, Hindustan Times and Indian Express starting from 17 January 2017 and till 21 March we have downloaded 26219 articles.
After that we are extracting the article's published date, title and content and storing in a relational database in following table.

ARTICLE:

ID	LINK	DATE	TITLE	TEXT	SOURCE

ID: Unique Identifier
LINK: Stores the link of article
DATE: Published Date
TITLE: Article's Title
TEXT: Stores content of article
SOURCE: News paper name
We have created another table to store person location and date details extracted after applying NER tagger.

EVENT EXTRACTED INFO:

ID	PERSONS	DATES	LOCATIONS	ARTICLE ID

ID: Unique Identifier
PERSONS: Comma separated value of all the persons extracted
DATES: Comma separated value of all the dates extracted
LOCATIONS: Comma separated value of all the locations extracted
ARTICLE ID: Foreign key to ARTICLE table
After processing this information the final result will be stored in following table

EVENT:

ID	TITLE	DATE	LOCATION	PERSON

ID: Unique Identifier
TITLE: Title of the event
DATE: Normalize date of event (DD-MM-YYYY)
LOCATION: Comma separated value of all the locations
PERSON: Comma separated value of all the persons/organization involved

4.2 Experiment Design

We developed a very primitive application for protest forecasting. As per functionality of our system, a person working on our system on a particular date can see the events of future from that date. While testing the system we will

pass the date as some date of Jan-March 2017 and get the events ahead of that day.

Implementation

1. Crawled articles and stored them in database and file.
2. Based on keys found using WordNet filtered files. We used cutoff value .68 to get the keys from this model.
3. Passed these files again to LDA for filtering more relevant document. We used following parameters for LDA: symmetric alpha: 0.0067, symmetric eta: 6.89, Topics: 50, 100, 150, Passes: 20
4. We then used Stanford NER Tagger and relation extractor to extract relation.
5. We used our designed algorithm to get relevant information.
6. Dates are normalized and results are stored in DB and file.
7. Web based UI developed where user can input the date. System will display protest events ongoing on that day or planned for future.

Evaluation

We designed evaluation of our system at each step. We evaluated our Event Related Article filtering model and Information Extraction Model. Then we evaluated our overall system.

1. We evaluated our Article filtering model on parameters precision, recall and f measure. To do so, we sample 250 articles from our main corpus and tagged them with relevance true (1) and false(0). Used standard formulas for calculation and arguments are calculated as below:
   True Positive (TP): Article is relevant and picked by our model
   False Positive (FP): Article is irrelevant and picked by our model
   True Positive (TN): Article is irrelevant and not picked by our model
   True Positive (FN): Article is relevant and not picked by our model.
2. Similarly, we evaluated our relevant information extraction algorithm modeling in terms of relevant relations. We tagged our relations for each article as relevant if they have relevant information. Each relation has been given an id and we stored relation id which were used for extracting information. We evaluated our model on precision, recall and accuracy parameters. TP, FP, TN, FN are calculated as previous, just instead of articles, we chose relations.
3. We evaluated our overall system using Precision@K parameters for correctness. We chose K to be 50.

5. RESULTS AND DISCUSSIONS

5.1 Relevant News Article Filtering

We started with application of LDA on our main corpus. Since number of protest related articles were very less, we were not getting good results. LDA was returning all the political articles which were noise for us.

To resolve this issue, we focused on improving desity of protest articles and started working on key phrase based filtering. Similar idea is also used in the paper [3]. We first tried synonyms but didn't get good results as we described earlier also. We used Word2Vec model to get similar words to PROTEST and DEMONSTRATION with cutoff 0.68.

```
Similar words to Protest
------------------------

('march', 0.8418266773223877), ('demonstration',
0.8105735182762146), ('protesting', 0.7690555453300476),
('protest,', 0.7672282457351685), ('agitation',
0.766545295715332), ('protests', 0.7635540962219238),
('demonstrations', 0.7546336650848389), ('stir',
0.7257044911384583), ('slogans', 0.7215386033058167),
('dharna', 0.7002922296524048), ('bandh', 0.6965096592903137),
('protest.', 0.6886583566665649)

Similar words to demonstration
------------------------------

('protest', 0.8132555484771729), ('march',
0.7872685194015503), ('vigil', 0.7572329044342041), ('rally',
0.728948712348938), ('demonstrations', 0.7267377972602844),
('agitation', 0.7242469787597656), ('staging',
0.7152520418167114), ('bandh', 0.7136560678482056),
('Protestors', 0.6897619962692261), ('dharna',
0.689063310623169), ('seminar', 0.6888107061386108),
('sit-in', 0.6877864599227905), ('stir', 0.6838892698287964)
```

Simply presence and absence of an keyword doesn't guarantee relevance. Form example:
Strike could relate to surgical strike or Russian strike on Ukraine which is not relevant to us. We applied LDA on this filtered corpus to get more relevant articles having details of protest event. We also wanted to rule our peripheral article which talk about protest but not actually related to protest event.

Result

Total Articles: 26,219
Articles Selected By Key Phrase Filtering: 825
LDA Filtered Articles: 357

Precision: 0.85
Recall: 0.695
F-Measure: .766

5.2 Information Extraction

We evaluate our information extraction model as described in section of Experiment Design. We got not so good results in our first attempt. There is window of improvement here.

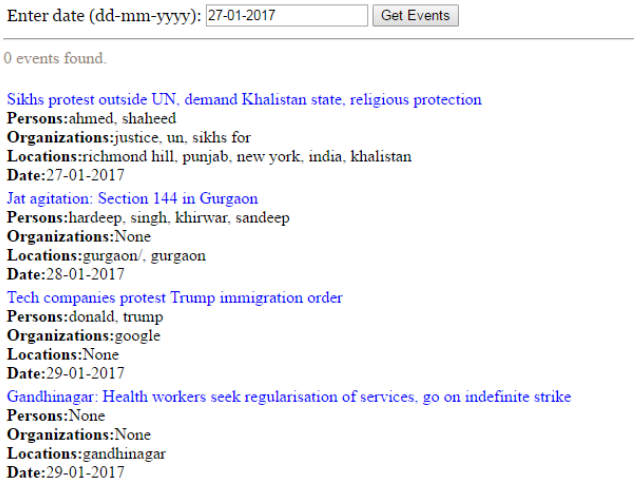
Result

Precision: 0.583
Recall: 0.63
Accuracy: .87

5.3 Final Results

After time expression normalization the results are stored in the database. We have built a web application which asks for date as input. Further we can extend using location based filters, ranking the protest in some way, may by defining intensity of event using sentiment analysis.

User Interface:



Evaluation Results:
Precision@50 : 0.66

6. CONTRIBUTION

We used proper project management technique and created detailed work item list along with ownership.

Neelesh K Shukla
- Problem Conceptualization and Design (Model, Implementation and Experiment)
- Implementation: Key Phrase Filtering (Word2Vec)
- LDA: Article Filtering using topics and cutoff
- Evaluation of event filtering module
- Name Entity Recognition Tagger and Relation Extraction
- Design of Model and Algorithm for Extracting Relevant Information
- Normalizing time expression and Web Based User Interface

Neha Saini
- Data collection Using Feed Parser
- Heuristic Building for Information Extraction
- Implementation of algorithm and heuristics for Extracting Relevant Information
- Evaluation of above implemented algorithm

Pranay Sanghvi
- Data collection and Extracting Article Title, Date and Text
- WordNet Implementation
- Heuristic Building for Information Extraction
- Normalizing time expression
- Worked on identifying nature of protest

Suweta Sakya
- Data Prepossessing (Tokenization, Lemmatization)
- Implementation: Key Phrase Filtering (SVD)
- Implantation: LDA (Doc-Topic and Topic-Word Mixture)
- LDA: Article Filtering using topics and cutoff
- Evaluation of event filtering module
- Report Creation

7. RELEVANT REFERENCES

1. Blei, David M.; Ng, Andrew Y.; Jordan, Michael I (January 2003). La_erty, John, ed. "Latent Dirichlet Allocation".
2. Stanford Named Entity Recognizer https://nlp.stanford.edu/software/CRF-NER.shtml
3. Muthiah, S; Huanbg, B; Arredondo, Jaime; Mares, D; Getoor, L; Katz, G; Ramakrishnan, 2015,N. 2015. Planned Protest Modeling in News and Social Media, , AAAI Publications.
4. Baeza-Yates, R. 2005. Searching the future. In SIGIR-Workshop on Mathematical/Formal Methods in Information Retrieval.
5. Llorens, H.; Derczynski, L.; Gaizauskas, R. J.; and Saquete, E. 2012. TIMEN: An open temporal expression normalisation resource. In LREC.
6. Jurafsky. D,; Martin. J,; Semantics with dense vectors http://web.stanford.edu/ jurafsky/slp3/16.pdf
7. Filannio M.; (Jun 2012) Temporal expression normalisation in natural language texts
8. WordNet, Princeton University; https://wordnet.princeton.edu/
9. Surdeanu, M; McClosky, D; Smith, M; Gusev, A; and Manning, Christopher D; 2011. Customizing an Information Extraction System to a New Domain. In Proceedings of the ACL 2011 Workshop on Relational Models of Semantics.