Event extraction from Social Media text using Conditional Random Fields

Nagesh Bhattu Sristy
IDRBT, Hyderabad
Hyderabad, Telangana
nageshbs@idrbt.ac.in

N. Satya Krishna*
IDRBT
Hyderabad, Telangana
satya.krishna.nunna@gmail.com

D. V. L. N. Somayajulu
NIT, Warangal
Warangal, Telangana
soma@nitw.ac.in

ABSTRACT

Social Media tools popularized the digital devices among masses making information dissemination easier and faster. Exchange of text is most popular effective means of communication across social media users. It has become necessity to process, understand the semantics of messages communicated as the messages have wide effect across the users. Event extraction refers to understanding the events across streams of social media messages. Event extraction helps in taking quicker corrective actions in case of natural calamities and hence possibly save lives of people. The main objective of the task is, drawing specific knowledge to predict the events(incidents) specified in digital text. We proposed two step procedure to extract events. First phase consists of applying a binary classifier to identify the messages, containing the event. Second phase consists of applying a sequence labeling technique, conditional random fields(CRF), to extract the event from the message. As social media text is noisy, it is a challenge to develop learning algorithms for these tasks. We use Parts of Speech (POS) tags of the words to address some of the issues in this challenge.

CCS CONCEPTS

• **Information systems → Structured text search**;

KEYWORDS

ACM proceedings, LaTeX, text tagging

1 INTRODUCTION

Twitter[1] and Facebook[2] messages provide up-to-date information about the current events. In todays world with proliferation in the usage of social media, extracting current events from unstructured tweets and posts has gained ample attention. Social media data consists unusual characteristics like short length, stylistic variations, acronyms, noisy and unstructured forms. This makes event extraction a challenging problem in Natural Language Processing (NLP). Numerous tools have been developed [14], [8] in the recent past for enabling short-text processing for various tasks such as POS Tagging, Chunking, Named Entity Recognition. These tools use special techniques to account for the out-of-vocabulary terms in text collected from twitter. Ritter et al. [14] uses brown's clustering on a huge corpus of english tweets to cluster similarly used words such as *'2morrow', 'tmrrow', '2mar'*. In comparison to english, the resources available for remaining languages are quite less even for formal text. This shared task pertains to processing text written in Indian languages namely *'hindi', 'tamil', 'malayalam'* to obtain the events from the social media messages.

Event Extraction is one of the most valuable tasks in Natural Language and Information Extraction. For example, accurate selection of news messages will improve the performance of news systems [5]. Furthermore, by detecting the occurrence of events, as early as possible, the performance of risk analysis systems Capet et al. [4], traffic monitoring systems Kamijo et al. [7] can be improved and forecasting civil unrest [13].

Early works, most of the methods Allan et al. [1] [6] Yang et al. [17] for event extraction have focused on news articles, which is the only best source of information for current events. With the ability of social media tools to virally popularize news items and their acceptance across masses, numerous media agencies have been relying on twitter, facebook feed pages to disseminate their news highlights. Twitter feeds for *hindi* [3,4], *tamil* [5,6] and *malayalam* [7,8] are few examples of social media forums continuously posting the news items. Among the posts made by these feeds, only a small fraction of tweets contain events.

Alen Ritter Allan et al. [1] developed the first open-domain event extraction tool (TWICAL) for twitter data. Extraction of NASDAQ-100 listed companies information from RSS feed using StockWatcher was proposed by [10]. Hermes Borsje et al. [3] is news interpreter that supports the decision making process to filter the relevant news using Semantic Web technologies. Using specific features related to the natural disasters, Sakaki et al. [15] proposed a method to detect the earthquake-related tweets. Benson et al. [2] presented a relation extractor to predict the artists and venues from tweets.

2 OVERVIEW

Social media text written in Indian languages has received much lesser attention compared to english. The multiplicity of languages in India and usage by comparatively lesser population can be thought of as the possible reasons for this observation. Tokenization is first step in processing the social media text written in Indian languages. Effective tokenization helps in segregating meaningful features from noise. Feature extraction involves conversion of text into lemma form and morphological analysis. POS tagging involves attributing POS tags for the text, observed as a sequence of words. We depend on the tools available [9] [12] for POS tagging of various Indian language sentences. The task is to identify event carrying

*Work done as part of Ph.D
[1]http://www.twitter.com
[2]http://www.facebook.com
[3]https://twitter.com/aajtak?lang=en
[4]https://twitter.com/bbchindi?lang=en
[5]https://twitter.com/news7tamil?lang=en
[6]https://twitter.com/thatstamil?lang=en
[7]https://twitter.com/manoramanews?lang=en
[8]https://twitter.com/beatsofkerala?lang=en
[9]http://ltrc.iiit.ac.in/download.php

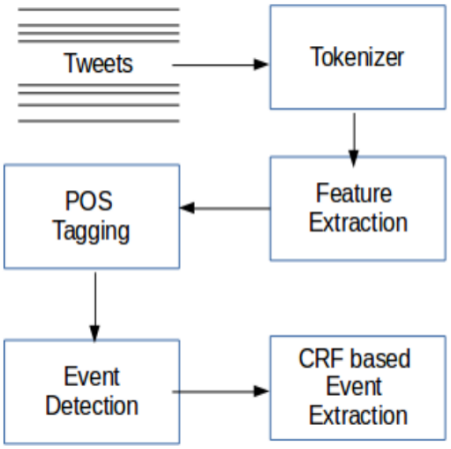

Figure 1: Overview of the approach followed

messages and then extract relevant portions of the events from such carrying text. The main phase of the approach consists of building a classifier to distinguish messages based on whether they carry event or not. Further post-processing is applied to extract the actual event from the event text. This is done through sequence tagger trained from the provided training data. Figure 1 indicates the overview of the our approach followed for the shared task. We use mallet [10] [9] the toolkit for the various machine learning algorithms applied in this work.

2.1 Problem Statement

Given a collection of sequence of words $\mathcal{D}$ where each sentence is of form $w_1, w_2 ... w_n$, identify the sequences carrying events. For the identified sequences a label sequence $l_1, l_2 ... l_n$ is predicted where each tag $l_i \in B, I, O$. The tags B,I,O indicate the beginning, inside and outside of event in event carrying text.

2.2 Tokenization & POS Tagging

Tokenization for our approach follows that of twokenizer [11]. This was extended to handle the unicodes of Indian languages. The indian languages *hindi,tamil,malayalam* have unicode ranges of 0x0900-0x097F, 0x0B80-0x0BFF, 0x0D00-0x0D7F respectively. Emoticons, urls, hashtags, userids are various other special tokens.

The POS taggers used from [12] have their own tokenizers and lemmatizers to do the morphological analysis. These tools are designed for news wire text which is supposed to be much more cleaner than the text seen on social media. We modified the respective tokenizers to get process the social media text.

[10] http://mallet.cs.umass.edu/index.php
[11] https://github.com/brendano/tweetmotif
[12] http://ltrc.iiit.ac.in/download.php

2.3 Event Detection

The critical component of our solution is the classifier which detects the events. We analysed the detection capabilities of 3 classifiers namely naive Bayes, logistic regression and semi-supervised naive Bayes EM. Naive Bayes (NB) algorithm models the classifier as an outcome of generative algorithm. If $\mathcal{D}$ is a training dataset of N examples $\mathbf{x_1}, \mathbf{x_2}, ...\mathbf{x_N}$ and $y_1, y_2 ... y_N$ are corresponding labels, NB model is expressed in the equation (1). We assume that the each $\mathbf{x_i} \in \mathcal{R}^F$ and each $y_i \in \{1, 2, .., L\}$ are the domains of respective portions of examples $(\mathbf{x_i}, y_i)$ where F is the number of features and L is number of labels. x_{ij} represents j'th features count in i'th example. w_j is the jth feature in the set of features. $p(w_j|y)$ is the j'th element of the parameter vector associated with label y.

$$maximize \sum_{i=1}^{N} log(p(\mathbf{x_i}, y_i)) \tag{1}$$

$$p(\mathbf{x_i}, y_i) = p(y_i)p(\mathbf{x_i}|y_i) \tag{2}$$

$$p(\mathbf{x_i}, y_i) = \prod_{j=1}^{F} p(w_j|y_i)^{x_{ij}} \tag{3}$$

Naive Bayes EM (NBEM) is a semi-supervised algorithm [11] which makes use of test data also along with the training data to infer the classifier. If the dataset training and testing portions of the dataset are designated as $\mathcal{D}_l$ and $\mathcal{D}_u$ respectively, the NBEM learns the classifier as a maximizer of (4). The first part of the objective is same as that of NB approach. Labels of test examples are not known and are learnt in an iterative EM algorithm, where E-Step predicts the labels of the test examples and M-Step learns the parameters of the model with the probabilistic labels learnt in the E-Step.

$$maximize \sum_{i \in \mathcal{D}_l} log(p(\mathbf{x_i}, y_i)) + \sum_{i \in \mathcal{D}_u} log(p(\mathbf{x_i}, y) \tag{4}$$

$$\tag{5}$$

Maximum entropy or logistic regression (MaxEnt) is a discriminative approach for building the classifier and hence does not get much benefit of the EM setting. MaxEnt learns the model which maximizes the objective in equation (6). The conditional distribution in equation (7) is softmax function. The numerator of equation (7) is the score of example $\mathbf{x_i}$ in class y_i. The denominator is normalizer which ensures that the value $p(y|\mathbf{x})$ is summing upto 1. It is known as maximum entropy, because Equation (6) is a dual of equivalent maximization of entropy under feature constraints. $\mu_\mathbf{y}$ is parameter vector corresponding to class y, which is of same size as that of F (number of features). L is the total number of labels.

$$maximize \sum_{i=1}^{N} log(p(y_i|x_i)) \tag{6}$$

$$p(y_i|\mathbf{x_i}) = \frac{exp({\mu_{\mathbf{y}_i}}^t . \mathbf{x_i})}{\sum_{y=1}^{L} {\mu_\mathbf{y}}^t . \mathbf{x_i})} \tag{7}$$

2.4 Event Extraction

Event extraction is performed using CRF [16]. If $(\mathbf{x_i}, \mathbf{y_i})$ is the i'th example where $\mathbf{x_i}$ is a feature sequence of lenght T and $\mathbf{y_i}$ is corresponding label sequence, the CRF models the maximization of joint conditional likelihood in Equation (6) where $p(\mathbf{y_i}|\mathbf{x_i})$ is defined as

Table 1: Dataset Characteristics

Language		No of Instances	No of Tokens	Dictionary Size
Hindi	Train	1025	19,497	4381
Hindi	Test	4451	89,167	11420
Malayalam	Train	2218	18449	4065
Malayalam	Test	5173	38625	13460
Tamil	Train	3843	37221	12033
Tamil	Test	5304	50365	15255

in Equation (8). The difference between the numerator in Equations (6) and (8) lies in features considered. CRF tries to model sequential dependencies, while MaxEnt classifier disregards sequential dependecies. The feature vector $\mathbf{f}(\mathbf{x_i}, \mathbf{y_i})$ in Equation (8) is similar to the feature vector in Equation (6), encoding number of times a feature associated with a label. The feature vector $\mathbf{g}(\mathbf{y_i})$ encodes the label sequence features or number of times a label combination appears in succession in the example $(\mathbf{x_i}, \mathbf{y_i})$. The number of features of $\mathbf{g}(\mathbf{y_i})$ vector is LXL each encoding possible label bigrams. The denominator $Z(\mathbf{x_i})$ is normalizer which is evaluated over all possible label assignments ($L^T for \mathbf{x_i}$)over label space. Evaluation of $Z(\mathbf{x_i})$ is efficiently done using forward-backward algorithm. μ and η are respective parameters associated with node features and edge features of CRF.

$$p(\mathbf{y_i}|\mathbf{x_i}) = \frac{exp(\mu^t.\mathbf{f}(\mathbf{x_i}, \mathbf{y_i}) + \eta^t \mathbf{g}(\mathbf{y_i}))}{Z(\mathbf{x_i})} \tag{8}$$

We used CRF++ [13] as our sequence labeller. CRF++ allows feature templates to be given for learning. The features being used for CRF++ are unigrams with a window of 5 words from the current word. As unigram features are extremely noisy as they are mostly seen very few times in the corpus and test data unigrams mostly are seen for the first time, we use POS tag based features also as second set of features to help the model inferred by CRF in mimicking event extraction process. We use similar 5 tag window for POS tags also. We used label based bigram features as label based features. The dataset provided for the shared task contains the starting ending positions of the event for each matched event text with in the text. We have converted this format of the input to the B, I, O based tagging to reflect the input for CRF++. As the tokenization employed adds spaces to reflect the tokenization process, the output of CRF is remapped to original to reflect the positions of the event as expected by the shared task.

3 EXPERIMENTS

The datasets taken for the task are summarized in Table 1. The preprocessing replaces all urls with keyword URL, *'@'* mentions are replaced with USER and hash-tags are all preserved as they sometime contain the event specific tags such as *#BombBlast #EarthQuake*. The test set for *hindi* is 4-times that of trainset while *tamil* and *malayalam* are relatively better in this ratio. Number of unique words is lesser for *malayalam*.

[13] https://taku910.github.io/crfpp/

Table 2: Event Detection Accuracy & F1 Score

Language	Method	Accuracy	F1-Score
Hindi	NB	73.66	0.6304
Hindi	MaxEnt	75.61	0.6641
Hindi	NBEM	80.00	0.7218
Hindi	MaxEnt + POS	80.0	0.7730
Hindi	MaxEnt + POS	80.0	0.7551
Hindi	NBEM	83.57	0.8096
Malayalam	NB	72.88	0.6243
Malayalam	MaxEnt	82.73	0.8583
Malayalam	NBEM	82.75	0.797
Tamil	NB	76.79	0.8437
Tamil	MaxEnt	77.05	0.8481
Tamil	NBEM	80.00	0.866

3.1 Event Detection Performance

We used mallet library for building the binary classifier for event detection. The training dataset consisted of event-text file and annotation file. The events text file consisted of one message for each line with additional details such as user-id, message-id. The annotation file shows message-id and event index for each event carrying message given in events file. We prepared a binary classifier treating the missing messages of annotation file as labelled 'no', while matched events as 'yes'. The performance of classifier is measured using two metrics namely Accuracy and F1-Score. They are defined in equations

$$Accuracy = \frac{NoofCorrectPredictions}{Totalnoof Predictions} \tag{9}$$

$$Precision = \frac{TruePositives}{TotalPositivePredictions} \tag{10}$$

$$Recall = \frac{TruePositives}{TotalPositiveexamples} \tag{11}$$

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{12}$$

We performed a 5 fold cross-validation to detect events. The classification accuracies and F1-Scores of different classifiers NB, ME, NBEM are reported in table 2

We can observe that MaxEnt classifier performs better than NB in all cases significantly, asserting the superiority of discriminative approaches. Adding POS tag features has improved the classification accuracy of MaxEnt and NB by 4.4% and 6.4% respectively. NBEM is the semi-supervised approach which is consistently better than the other two methods with and with-out POS tags, as it uses the test portion of the data for learning its model.

3.2 Event Extraction Accuracy

The event extraction module contains the tags B, I, O indicating the beginning, inside, ending of a event. The accuracy and F1-Score of Equations (9) and (12) are extended for the CRF output and the tagging effectiveness is reported in table 3. Our submission only includes only one language namely *hindi*.

Table 3: Event Extraction Accuracy & F1 Score

Language	Precision	Recall	F-measure
Hindi	31.56	71.39	43.77

REFERENCES

[1] James Allan, Ron Papka, and Victor Lavrenko. 1998. On-line new event detection and tracking. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 37–45.

[2] Edward Benson, Aria Haghighi, and Regina Barzilay. 2011. Event discovery in social media feeds. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 389–398.

[3] Jethro Borsje, Leonard Levering, and Flavius Frasincar. 2008. Hermes: a semantic web-based news decision support system. In *Proceedings of the 2008 ACM symposium on Applied computing*. ACM, 2415–2420.

[4] Philippe Capet, Thomas Delavallade, Takuya Nakamura, Agnes Sandor, Cedric Tarsitano, and Stavroula Voyatzi. 2008. *A Risk Assessment System with Automatic Extraction of Event Types*. Springer US, Boston, MA, 220–229. https://doi.org/10.1007/978-0-387-87685-6_27

[5] Philipp Cimiano and Steffen Staab. 2004. Learning by Googling. *SIGKDD Explor. Newsl.* 6, 2 (Dec. 2004), 24–33. https://doi.org/10.1145/1046456.1046460

[6] George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. 2004. The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation.. In *LREC*, Vol. 2. 837–840.

[7] Shunsuke Kamijo, Yasuyuki Matsushita, Katsushi Ikeuchi, and Masao Sakauchi. 2000. Traffic monitoring and accident detection at intersections. *IEEE transactions on Intelligent transportation systems* 1, 2 (2000), 108–118.

[8] Chenliang Li, Jianshu Weng, Qi He, Yuxia Yao, Anwitaman Datta, Aixin Sun, and Bu-Sung Lee. 2012. TwiNER: Named Entity Recognition in Targeted Twitter Stream. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '12)*. ACM, New York, NY, USA, 721–730. https://doi.org/10.1145/2348283.2348380

[9] Andrew Kachites McCallum. 2002. MALLET: A Machine Learning for Language Toolkit. (2002). http://mallet.cs.umass.edu.

[10] Alex Micu, Laurens Mast, Viorel Milea, Flavius Frasincar, and Uzay Kaymak. 2009. Financial news analysis using a semantic web approach. In *Semantic Knowledge Management: An Ontology-Based Framework*. IGI Global, 311–328.

[11] Kamal Nigam, Andrew Kachites Mccallum, Sebastian Thrun, and Tom Mitchell. 2000. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning* 39, 2 (01 May 2000), 103–134. https://doi.org/10.1023/A:1007692713085

[12] Avinesh PVS and G Karthik. 2007. Part-of-speech tagging and chunking using conditional random fields and transformation based learning. In *Proceedings of IJCAI Workshop on Shallow Parsing for South Asian Languages*, Vol. 21. 21–25.

[13] Naren Ramakrishnan, Patrick Butler, Sathappan Muthiah, Nathan Self, Rupinder Khandpur, Parang Saraf, Wei Wang, Jose Cadena, Anil Vullikanti, Gizem Korkmaz, Chris Kuhlman, Achla Marathe, Liang Zhao, Ting Hua, Feng Chen, Chang Tien Lu, Bert Huang, Aravind Srinivasan, Khoa Trinh, Lise Getoor, Graham Katz, Andy Doyle, Chris Ackermann, Ilya Zavorin, Jim Ford, Kristen Summers, Youssef Fayed, Jaime Arredondo, Dipak Gupta, and David Mares. 2014. 'Beating the News' with EMBERS: Forecasting Civil Unrest Using Open Source Indicators. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*. ACM, New York, NY, USA, 1799–1808. https://doi.org/10.1145/2623330.2623373

[14] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named Entity Recognition in Tweets: An Experimental Study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 1524–1534. http://dl.acm.org/citation.cfm?id=2145432.2145595

[15] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*. ACM, 851–860.

[16] Charles Sutton, Andrew McCallum, et al. 2012. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning* 4, 4 (2012), 267–373.

[17] Yiming Yang, Tom Pierce, and Jaime Carbonell. 1998. A study of retrospective and on-line event detection. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 28–36.