International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Event Extraction And Classification From English Articles

Vanitha Guda, Y.RamaDevi

Abstract*: *Today's digital world huge number of information sources like wikis, web, blogs and other sources are creating a lot of information with several events. Basically, an event can be a situation, action or state that can be represented in natural language text in the form of happening or occurrence. Analyzing the event information finding the relation between the events is one of the crucial tasks in information retrieval. In a formal way, the event can be defined as a real-world entity that happens or occur; these are the dynamic occurrences which have causes or effects (E.g. earthquake, floods, crime, etc.). Extracting events, events fall within a timelines extraction can be applied in many of the natural language applications like text summarization, temporal question answering systems, etc. Event extraction and classification can use in other related text searches like News domains, legal documents, wikis, manuscripts, and time-based searches. In this paper, we present a methodology for event extraction in natural language text which helps in finding out the type of an event and classifies the events under specific categories. Our work aims to develop a system which would automatically identify events from articles generated over the internet. The system would not only detect the events but also tried to detect important times of the event. Finally compared the accuracy of work with several classifiers and obtained results shows good accuracy measure for Support Vectors machine (SVM).

IndexTerms*: *Natural language Processing, Events Extraction, Events, Time, and Classifiers.

I. INTRODUCTION

Nowadays in this digital era communication has become very fast and easy with that proportionally the amount of data is increasing. There are 2.5 quintillion bytes of data created each day at our current pace, but that pace is only accelerating with the growth of IOT (Internet of Things). Over the last two years alone 90 percent of the data in the world was generated, most of the content is in the form of text. According to the worldwide web (WWW) survey from 2013, the number of Tweets generated for each minute has increased 58% to more than 455,000 Tweets per minute in the year 2018. Instagram users upload 46,740 million posts every minute. From the time 2013, the count of Facebook Posts shared each minute has increased by 22%, from 2.5 Million to 3 million posts per minute in 2016. Every year this number has been increased more than 300 percent, from around 600,000 posts per minute in 2011. According to worldometers almost 3 to 4 million articles or posts written every day and there is a serious necessity to analyze the data.

An event that happens in any part of the world gets communicated in a few seconds or minutes to the rest of the world. For example, the recent bomb blast in Syria was known to the world within a few minutes through media. There is a great need to automatically identify various events like bomb blasts, floods, cyclone, fires, political any kind of events, etc., reported in various newswires, Social Media text. The task is to identify various events and their span such as sports events, terrorist events, natural disasters, crime events, corporate events, political events, accidents, etc, in a given text. Further going ahead in this task, along with the identification of event and times which are related to an event identification and classification of the event for further retrieval is the vital task. The actual real-time applications will be benefited only if the full information related to the event is identified.In this Paper, we present a methodology for the event extraction and classification of the event with the category, Event-Time (E-T) Relations from the given natural language text articles. In section-II brief literature about the related works, section-III describes the workflow and implementation detail, Section-IV presents Datasets and Results, observations and finally Section-V concludes the paper.

II. THE LITERATURE OF RELATED WORKS

A. Review Stage

Most of the related works presented in two categories one is domain dependent and other is domain independent of events extraction [1] where domain independent categorizes the events based on lexical features. Instead of analyzing the data most of the researchers focused for the data which changes over the time. Focusing on the hidden patterns of information within period or certain time slot for the better way of analysis. One major note is most of the recent works in this context concentrate on Event detection in English text, very few of them involves the Chinese- language also.With the reference of Borsje[2] change mining is new way of generation in data mining which means changes for data and it is necessary to detect the changes in the procedures. To consider the Change Mining method it includes specification, change in time and detection mechanism and change modeling.

Revised Manuscript Received on 30 July 2019.
***** Correspondence Author
Vanitha Guda*, CSE Department, Chaithanya Bharathi Institute of engineering Technology(A), Gandipet, Hyderabad, India.
Prof Dr Y.Rama Devi, CSE Department, Chaithanya Bharathi Institute of engineering Technology(A), Gandipet, Hyderabad, India.

In [3] also focused on detecting changes in the pattern. Change mining is one type of the mining in high order for additional technologies such as clustering, trend analysis and classification. Event can be represented with several features that help to enhance the event extraction process.

According to Chambers [4], for the time temporal and hierarchical way of event detection can claim that the time is the major dimension for event detection. The features should be extracted based on the burst time feature and used to find the related documents. According to F. Hogenboom [5] the model for Over the topics over Time (TOT), it is the extension for Latent Dirichlet Allocation (LDA), model it explains how to consider the time attribute. Using LDA, it avoids discretization by assigning each topic a continuous distribution over time. The performance of the TOT[11] is better if compared with the LDA. The main limitations are in the time sequence, knowledge of the events [6] are extracted is in a static way. But it is difficult to consider the events for a specific topic. It is also one of our contributions in this paper for a given topic, we aim to extract the events, identifying the events fall within time and classifying the events with specific categories.

III. WORK FLOW AND IMPLEMENTATION

3.1ModulesDescription: This section refers to the modules of workflow shown in Figure-1

Multiple documents: These are the documents which we use as input to train our model. These documents are taken from FIRE [9] and online news articles. The documents consist of the information related to news article; news articles can have more number of events it is easy to analyze what type of event has occurred and the number of casualties of the event.

Data Preprocessing: This is the most important phase in our project. This phase helps in understanding the data better and also helps in the accurate calculation in TF-IDF.

The steps in the data preprocessing module are:

i. Removing special characters
ii. Removing accented text
iii. Tokenization
iv. Stop words removal
v. Stemming

i. Removing Stop Word list: A stop list consists of all the stop words that are highly used by researchers in NLP processing techniques. This list of stop words contain mostly all the words which have higher significance as a stop word.

ii. Stem list: A stem list consists of all the words stem forms. Basically, it has all the words root form. Eg: Words am/are/is converted in be. This says the root word of being, is, are be. This step helps in increasing the specificity of each word and removes the ambiguous meaning of different words.

iii. Term Frequency and Inverse Document Frequency (TF-IDF):

Term Frequency: It is the number of times a word or term occurs in the document.

Document Frequency: It is the number of documents in which a word or term exists.

Inverse Document Frequency: It is the inverse of Document Frequency multiplied by a number of documents.

iv. **Classification of Event:** It is the phase where the data is fed to the model and the classification of the event occurs. The event to which the document belongs to is classified. Based on the number of correct classification and false classification a confusion matrix is created. We get to know the class of event to which the document belongs to.

v**. Extraction of dates:** The step extraction of dates extracts the important dates and times in the document. For example, let's say, an event occurred on Monday. This step extracts the date or time [10] component Monday and adds it to the important date or time components in the document.

vi. **Accuracy metrics:** This is the last step of the process [11] accuracy metrics of the model are calculated and are evaluated. There are Different techniques to calculate accuracy metrics like Precision, Recall, and F-measure etc. We will discuss each of the techniques clearly in the implementation phase.

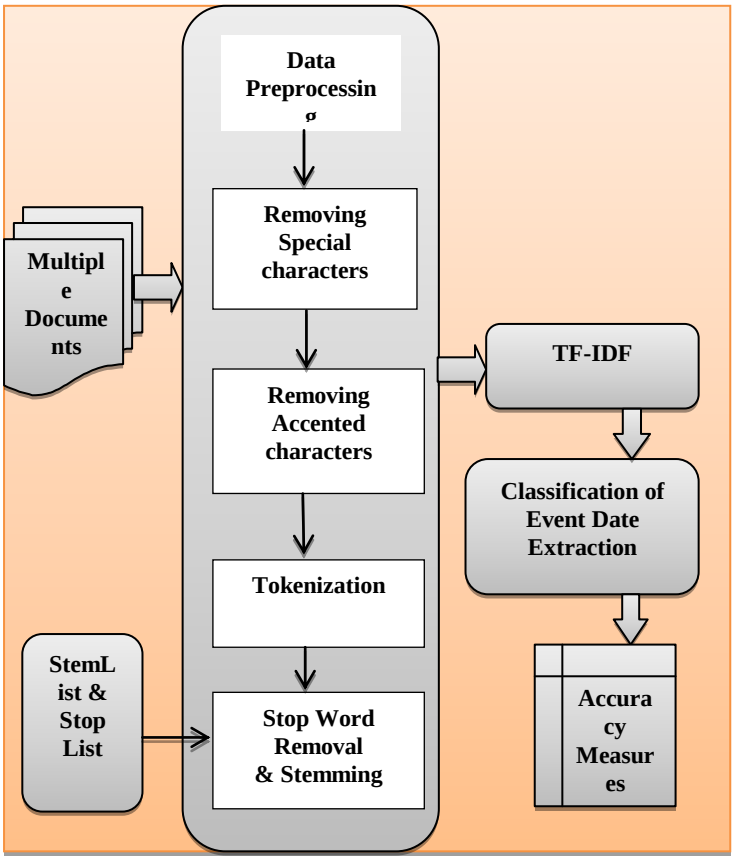

Figure-1, Workflow of the Proposed system

3.2 Implementation Details of the proposed system

- **Punctuation and special characters Removals**: The symbols are specific characters are non-alphanumeric characters or numeric which may add extra noise to the text. Regular expressions

- (Regexes) can be used to remove the punctuations and special characters.
- Punctuation string: &\'()*+,-./:;<=>?@[\\]^_`{|}~'

We shall loop through every character in the document and remove any punctuation special characters.

- **Accented Characters Removal:** generally in any corpus, dealing with accented characters or letters, if you only want to analyze the English language. Hence, we need to make sure that these characters are converted into standardized ASCII characters. A simple example: converting é to e et,.
- **Expanding Contractions:** Contractions are a shortened version of words or syllables. In English syllables are exist in written or spoken forms. The shortened contractions of words may create by removing any specific letters and sounds. In the case of English contractions, they are often created by removing one of the vowels from the word. Examples would be, do not to don't and I would to I'd. Each converted contraction helps for text standardization.
- **Tokenizing:** Tokenization describes splitting paragraphs into sentences or sentences into individual words. Each Sentence can be split into individual words and punctuation most commonly this split across white spaces. Tokenizer Segments text into words. To do this tokenizer loops through every sentence and splits it into words by space. While tokenizing we also make all the characters lowercase.
- **Removing Stop words:** The words with no significance or less specific, while forming the meaningful features from the text, are known as stop words. These words that end up will have the maximum frequency of terms or word frequency. Words can be articles, conjunctions, prepositions and so on. Examples of stop words are a, an, the, and the like.
- **Lemmatization:** Lemmatization is where we remove word affixes to get to the base form of a word. The base form is known as the root word, but not the root stem. The difference is that the root word is always a lexicographically correct word (present in the dictionary), but the root stem may not be so. Lemma is root word known as the lemma, will always be present in the dictionary.
- **Feature Extraction:** Feature Extraction and Representation is one of the important tasks in NLP Documents or Words are converted and represented in vectors.

A. Two methods:

i. Bag of Words Model (Uni-Gram, Bi-gram, Tri-Gram).
ii. TF – IDF

Bag of words Unigram Model: This method to extract features from text documents [6]. These features can be used for training machine learning algorithms. It creates a vocabulary of all the unique words occurring in all the documents in the training set. For example, if you have 3 documents

D1 - I am feeling very happy today"
D2 - "I am not well today"
D3 - "I wish I could go to play"

Figure-2, bigram Model

First step: Performs a vocabulary using unique words from all the documents unique list of words here: "I am feeling very happy today not well wish could go to play" Then D1, D2, and D3 can be represented as Figure-2, shows the vector table for a Unigram Model. As we can see the words are divided into individual units as per the Unigram model.

1) Bag of N-grams: Bi-Gram, Tri-Gram Model :

- A unigram is nothing but a single word [6].
- We already know that bag of words consider the order of words
- What if we wanted it to account for the sequence of words or phrase
- That's where bi-grams and tri-grams come into place

Two different words are combined as per the Bigram model, using this reference it is easy to create vectors for Tri-gram, N-gram, etc.

- TF-IDF (Term Frequency-Inverse Document Frequency)

There are some potential problems which might arise with the Bag of Words model when it is used on large corpora. Since the feature vectors are based on absolute term frequencies, there might be some terms which occur frequently across all documents and these may tend to overshadow other terms in the feature set.

The TF-IDF model tries to combat this issue by using a scaling or normalizing factor in its computation.TF-IDF as stated in [12] is a combination of TF (Term Frequency) and IDF (Inverse Document Frequency). Term Frequency: It is the number of times a word or term occurs in the document.

TF(t,d) = Σf(t, ti)i=1 to n where n is the number of words in the document.

Σf(t,ti) = 1 if t = ti
Else
Σf(t,ti) =0.Logarithmically scaled TF becomes,TF2(t,d) = log(TF(t,d)+1)

- **Document Frequency**: It is the number of documents in which a word or term exists. DF(t) = Σg(t,di) , i=1 to N , where N is the number of documents. Σg(t,di) = 1 if dicontains tElse Σg(t,di) =0
- **Inverse Document Frequency:** It is the inverse of Document Frequency multiplied by number of documents. IDF(t) = log (N/ DF(t)).TF-IDF is the product of TF and IDFTFIDF(t,d) = TF(t,d) x IDF(t).

The total number of document in corpus (N) is constant, IDF mainly depends on the DF. The words which are having the similar DF value will be with the same IDF value. The TF-IDF calculations are simple and straightforward, and it does not require meaning or specific knowledge. These approaches helps to find the inter-document or intra-document statistical structure.

- Extracting important date or time components in the document:

Algorithm: Step 1: start
Step 2: After the preprocessing of data is done, we give the document to date or time prediction.

	I	am	feeling	very	happy	today	not	well	wish	could	go	to	play
D1	1	1	1	1	1	1	0	0	0	0	0	0	0
D2	1	1	0	0	0	1	1	1	0	0	0	0	0
D3	2	0	0	0	0	0	0	0	1	1	1	1	1

Step 3: We iterate the words in the document and check the label of each word. If the label is DATE or TIME, We add the component which is responsible for such label to the array of date or time components.

Else Ignore the word and proceed to the next word.

Step 4: Repeat Step 2 until the end of the document.

Step 5: End.

- **Calculating Model accuracies:** After the classification of the model is done create a confusion matrix to calculate accuracies.
- **Confusion matrix:** A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. Most performance measures are computed from the confusion matrix. confusion matrix Structure.

Table-1 Demonstrating a confusion matrix

	Predicted Positive	**Predicted Negative**
Actual True	**TP**	**TN**
Actual False	**FP**	**FN**

IV. DATA SETS AND RESULTS

Our dataset is a set of documents. Each document consisting of a description about an event occurred and its details. This dataset is taken from FIRE (Forum for information retrieval) [9]. It consists of various semi-structured documents (XML) consisting of events related to11 categories. The categories our project is emphasized on are Accidents, Crime, Cyclone, Earthquake, Fire, Floods, Shootout, Storm, Suicide Attack, Volcano.

We used a python script to change the documents from XML format and strip them into a format suitable for our model. Basically, the script checks for any XML tags in each document and removes the tags from our document[7]. Thus the final text in the document will be plain text without any XML tags.

Description: After training the Model with FIRE[4] dataset we faced an issue with the amount of data. This insufficient data isn't helping to train the model accurately. Thus we started scraping news articles from news wires and social media. We created a new data set consisting of the same 11 categories.

4.1 Results:

Before presenting the results the testing process we will discuss, we feed the model with our test split documents. The model predicts the appropriate class of the model. Based on the predicted class and actual class of the document the confusion matrix is created. We discussed the creation of confusion matrix and the terminologies involved in it in the Algorithm step. We calculate metrics after the confusion matrix is created.

Models Used and Accuracy Metrics:

Table-2, shows the Accuracy, Precision, Recall and F-score of Classifiers.

Table-2, shows the Accuracy, Precision, Recall and F-score of Random Forest Classifier, Multinomial Naive Bayes, Logistic Regression and Linear SVC

Model Name	Random Forest Classifier	Multinomial Naive Bayes Classifiers	Logistic Regression	Linear SVC
Accuracy	53%	55%	68%	90%
Precision	31%	45%	65%	88%
Recall	27%	31%	45%	81%
F-score	26%	33%	49%	86%

- **Accuracy Scores on 5-fold Cross Validation**

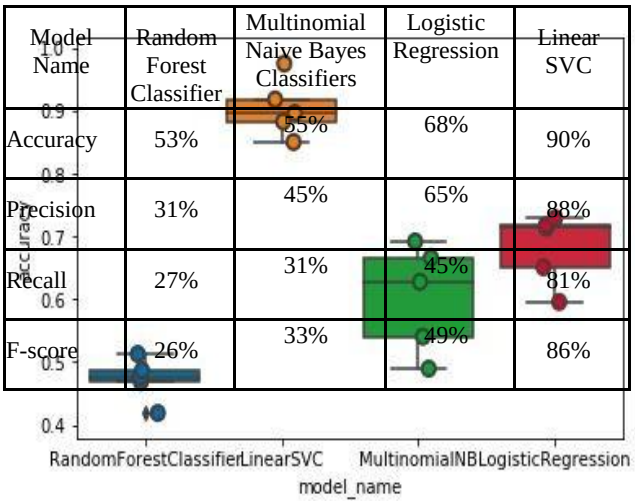

Figure 3: Accuracy Scores on 5-fold Cross Validation

Figure -3 presents Accuracy plots of models of Random Forest Classifier, Linear SVC, Multinomial Naive Bayes and Logistic Regression.

Description: Each round circle shown in the model is a split of dataset used as shown in Figure-3. We did 5-fold cross validation. The lowest horizontal line is the least accuracy prediction. The highest horizontal line is the highest accuracy prediction. The shaded area depicts the average accuracy of each model. As the accuracy plot depicts the higher performing model is Linear SVC and the least performing model is Random *Forest Classifier*.

Table-3, Accuracy values of classifiers

Model	**Accuracy**
Random Forest	**47%**
Logistic Regression	**60%**
Multinomial NB	**68%**
Linear SVC	**90%**

Table-3,Presents the accuracy percentage of the models: Random Forest Classifier, Linear SVC, Multinomial Naive Bayes and Logistic Regression. Random Forest performed the least..

- **Performance of Models:**

Random Forests perform slightly worse in the following cases: When the dimensionality is very high with respect to the number of training samples. They fail in sharp corners and exactness. They use diffusion methods. They do not fit elaborate and highly detailed things well when the sample size is low.

Logistic Regression and Multinomial Naive Bayes models are traditional models and they perform modestly better over Random Forests in NLP [8] but very much far away from SVC's. Support Vector Machines outperform all the other models by a large margin mainly because: Firstly it has a regularization parameter, which makes the user think about avoiding over fitting. Secondly it uses the kernel trick, so you can build in expert knowledge about the problem via engineering the kernel. Thirdly an SVM is defined by a convex optimization problem (no local minima)for which there are efficient methods (e.g. SMO). Lastly, it is an approximation to a bound on the test error rate, and there is a substantial body of theory behind it which suggests it should be a good idea..

V. CONCLUSION

We have successfully classified events with effective pre-processing techniques and model obtained 90% of accuracy in events extraction, with increasing amount of data the accuracy can be further improved. The Linear SVC model performed better than all other models. We also understood the merits and demerits of various models used in our work. Further with more data we can try deep learning models which work very well on huge amounts of data. This model can be deployed in any real time application to help readers to get articles based on the categories..

REFERENCES

1. Aone, C., Ramos-Santacruz, M.: REES: A Large-Scale Relation and Event ExtractionSystem. In: 6th Applied Natural Language Processing Conference (ANLP2000). pp.76-83. Association for Computational Linguistics (2000).
2. Borsje, J., Hogenboom, F., Frasincar, F.: Semi-Automatic Financial Events DiscoveryBased on Lexico-Semantic Patterns. International Journal of Web Engineeringand Technology 6(2), 115{140 (2010).
3. "Dr.S.Kannan ; VairaprakashGuruswamy (2015) Preprocessing Techniques for Text Mining""Riddhi Dave, PremBalani; Different Lemmatization Approches 2015"
4. Chambers, N., S. Wang, and D. Jurafsky. 2007. ClassifyingTemporal Relations between Events. In Proceedings of theACL 2007 Demo and Poster Sessions, pages 173–176,Prague, Czech Republic, June."An improvement of TF-IDF weighting in text categorization Mingyong Liu1+ and Jiangang Yang" ICCTS 2012.
5. F. Hogenboom, F. Frasincar, U. Kaymak, and F. De Jong. An overview of event extraction from text. In Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011) at Tenth International Semantic Web Conference (ISWC 2011), volume 779, pages 48–57, 2011
6. Amit Singhal. Modern Information Retrieval: A Brief Overview. IEEE, 2001
7. C. Shang, A. Panangadan, and V. K. Prasanna. Event extraction from unstructured text data. In International Conference on Database and Expert Systems Applications, pages 543–557. Springer, 2015.
8. Benson, E., Haghighi, A., Barzilay, R.: Event discovery in social media feeds. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. (2011)
9. www.fire2019.com.
10. Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Huan Liu and PhilipS. Yu, "Time-dependent event hierarchy construction", Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, San Jose, California, USA, pp. 300-309, 2007
11. Xuerui Wang and Andrew McCallum, "Topics over time: anon-Markov continuous-time model of topical trends",Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, Philadelphia, PA, USA, 2006.
12. GooLimin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. Structured relation discovery using generative models. In Proceedings ofthe 2011 Conference on Empirical Methods in Natural Language Process-ing, pages 1456{1466, Edinburgh, Scotland, UK., July 2011. Associationfor Computational Linguistics.
13. Roman Yangarber, Winston Lin, and Ralph Grishman. Unsupervised learning of generalized names. In Proceedings of the 19th InternationalConference on Computational Linguistics (COLING 2002)gleCaffeine, http://www2.sandbox.google.com.

AUTHORS PROFILE

Vanitha Guda Assistant professor at Chaithanya Bharathi Institute Of Technology Hyderabad. She is the Life member of Computer Society India-CSI, ISTE. Her Research interests are Natural Language Processing ,Question Answering and published 15 papers in International, National journals and conferences.

Dr Y.Rama Devi, professor in CSE Department, Chaithanya Bharathi Institute Of Technology Hyderabad. She published nearly 50journals, 57 conferences publications in international and national conferences. She is the life member of, ISTE,IETE, IEEE, IAENG, IACSIT and CSI.