

Event Time Relationship in Natural Language Text

<https://doi.org/10.3991/ijes.v7i3.10985>

Vanitha Guda ^(✉)

Chaithanya Bharathi Institute of Technology (A), Hyderabad, India
gvanitha_cse@cbit.ac.in

SureshKumar Sanampudi

JNTUH College of Engg. Jagityal, Karimnagar, India

Abstract—Due to the numerous information needs, retrieval of events from a given natural language text is inevitable. In natural language processing (NLP) perspective, "Events" are situations, occurrences, real-world entities or facts. Extraction of events and arranging them on a timeline is helpful in various NLP application like building the summary of news articles, processing health records, and Question Answering System (QA) systems. This paper presents a framework for identifying the events and times from a given document and representing them using a graph data structure. As a result, a graph is derived to show event-time relationships in the given text. Events form the nodes in a graph, and edges represent the temporal relations among the nodes. Time of an event occurrence exists in two forms namely qualitative (like before, after, duringetc) and quantitative (exact time points/periods). To build the event-time-event structure quantitative time is normalized to qualitative form. Thus obtained temporal information is used to label the edges among the events. Data set released in the shared task EvTExtract of (Forum for Information Retrieval Extraction) FIRE 2018 conference is identified to evaluate the framework. Precision and recall are used as evaluation metrics to access the performance of the proposed framework with other methods mentioned in state of the art with 85% of accuracy and 90% of precision.

Keywords—Natural language processing, Temporal Events, Event-Time Graph, Question Answering, Time Graph.

1 Introduction

In the present digital information era, there is exploitation of data which is resulting in a massive number of textual information sources (e.g., web, majorly news, tweets, historical texts, electronic health records, legal reports) that are mostly with the descriptions of events, extracting and analyzing events from a given document is found to be an essential task. In linguistic terms events in the text are referred to as event mentions. Due to ambiguities and vagueness of text representation in natural language, the mapping of real-world events and their relations counterparts causes some loss of information.

Existing works of NLP mainly discuss the sentence-level event mentions and document level events within the topic. In Topic Detection and Tracking (TDT), "event"

is referred to be something that occurs at a specific place and time whereas the topic is a set of news stories strongly related by some real-world event as stated in [1]. To group the news stories by themes, TDT has mostly relied on the traditional vector space model as stated in [2]. More recent approaches extend the vector space model by particular entity classes (e.g., named entities, noun phrases, collocations, among which named entities are found to be important because they often identify the participants of an event as stated in [3] [4] [5]).

Event-oriented information needs to involve structured queries rather than keyword-based queries (e.g., ‘What are the countries that Prime Minister Narendra Modi has visited and in which of his visit he declared protection laws?’). This type of query representation can be beneficial for temporal question answering (QA) systems as stated in [6] e.g., ‘who won both the Sidney and New York marathons three years in a row?’). The QA system results in multiple responses, where the gap of missing and incorrect information is filled by merging the data obtained from multiple sources.

This paper aims to build a framework for the extraction of events and times and represent the event-time relations using a graph data structure. An event graph is a labeled graph in which nodes represent individual event mentions, and edges represent temporal relations between event mentions. The initial stage of framework extracts event-time relations at sentence-level events, later event-time relations are inferred for document-level events.

The main contribution presented in the paper talks about the development of a framework for constructing event-time graphs to a given natural language text. Although graph-based event-time modeling is language-independent, this paper focuses on the processing of English language texts, for which state-of-the-art methods are compared to evaluate the performance. Constructing event-time graphs from the raw text is observed to be a challenging task due to the following reasons.

- To integrate machine learning and rule-based methods for various tasks like dataset annotation, model construction, and evaluation.
- To pipeline the activities (e.g., for parsing, named entity recognition, entity co-reference resolution) of the NLP to design the framework for building event-time graphs.

Precision, recall and F-Score are used as standard evaluation metrics to evaluate the performance of NLP works.

The continuing part of the paper is structured as follows. Section 2, talks about the related work of event extraction and time extraction from a given text. Section 3, explains the workflow of the proposed framework. Section 4 elaborates the process of building the event-time graph. Section 5 illustrates the corpora used for experimentation. Section 6 presents the evaluation of outputs obtained for the proposed framework. Finally, Section 7 outline the conclusions and future scope of work.

2 Related Work

Events and Time play a vital role in NLP applications like text summarization, question answering systems. The available literature talks about the extraction of events and times as independent units in the system. A brief overview of literature about event extraction explained in Section 2.1 and time extraction in Section 2.2.

2.1 Event extraction

The concepts of event extraction as the task started as part of the Message Understanding Conferences (MUC) as stated in [7], where templates of events are defined for specific domains. The templates were predetermined, and the task was to identify the events and classify them. Early approaches and systems for extracting event-based information were rule-based as stated in [8]. For each domain, new extraction patterns are designed as stated in [9]. The recent approaches focus on acquiring templates automatically, from the study conducted on event extraction are those works can be categorized into three classes

- Data-driven method
- Knowledge-driven method
- Hybrid methods.

Data-Driven approaches: Data-driven approach generates the rule for event extraction. Quantitative methods and lexical features are used to derive template based rules. It encompasses all quantitative approaches to automate language processing. As stated in TempEval-2013[10] (Temporal Evaluation), Zhen [11], STEP [12], explain the methods and features used to generate the rules for extracting the events. Though these methods are simple to implement; they do not consider semantics during rule construction. Rules are not exhaustive, need to build a new rule when a new situation arises. All the methods used in this class are non-knowledge based.

Knowledge-driven: This type of method uses knowledge about the content and encodes them into patterns. For the generation of a pattern, rules are constructed using lexico-syntactic or lexico-semantic features. Lexico-syntactic exploits use syntactical information. Lexico-Semantic uses the meaning of the information along with lexical representations. As stated in REES [9], EVITA [13] explain the methods which make use of syntactic and semantic information in building the event-patterns. Though pattern-based approaches have the advantage of using very less amount of data, patterns definition require prior domain knowledge. Designing and maintaining patterns is difficult when patterns need to be scaled-up to cover more situations. The feasibility and accuracy of rules depending on the user's knowledge of linguistics and domain expertise.

The hybrid method: This method takes the advantages of data-driven and knowledge-driven methods. Besides, hybrid methods use machine learning approaches. In this method, expert knowledge is applied to the output of a statistical model to prune unwanted results or to include information that could have been missed by

statistical models. Also, researchers have combined statistical approaches with (lexical) knowledge, where encoded lexical knowledge forms the features for supervised learning. As stated in FrameNet[14] or VerbNet [14], methods are built using hybrid methods. In these methods, complexity increases due to increasing of data and the combination of the multiple techniques. Expert knowledge is needed for efficient event discovery because most of these systems are domain specific.

2.2 Time extraction

Time is a temporal expression that indicates the point or period of happening in a situation. The representation of time exists in two forms, Quantitative form (15th July 1985, 15/07/1985) and Qualitative form (After the break, during the war). The works for time extraction can be categorized into rule-based and annotation systems.

Rule-based systems: Rule-Based systems build templates on labeled corpora such as [15] these templates recognize quantitative times which are calendric times. Later GUTime: a Perl temporal tagger provided by Georgetown University which is based on TempEx consisting of components for the extraction of events, temporal expressions, and temporal relations for recognizing and normalizing temporal expressions in English text implemented for the clinical narratives. To extract and normalize time expressions detection and normalization Temporal tagger it was adapted from publicly available Temporal Tagger. Next is BIO classifier[31] tags each input token as either Beginning, In, or Out of a time expression with nine tags. Temporal Expression Recognition as a BIO token-chunking task, where each token in the text is classified as being at the B(eginning) of, I(nside) of, or O(utside) of time expression. A rule-based system is only for the Quantitative times and Domain Specific.

Annotation based systems: Annotation for time expression analyzes the ISO times and builds mark-up language is a Markup or representation language used to capture temporal information but it can only represent the language cannot perform reasoning of the generated events with times. For annotating event and temporal information in text, the standards benchmarks are as stated in TimeML [21], and for the datasets, as stated in TimeBank [20] quoted in the literature. It has the transition from template-based event extraction to more robust and general approaches. Early work on event extraction, ACE [22] focused on domain-specific event extraction; the tasks included the extraction of event anchors, arguments and event co-reference resolution. TempEval campaign as stated in [23] focused on the extraction of relations between events and temporal expressions, extraction of events and document creation time, identification of main events in adjacent sentences. The second TempEval campaign as stated in [21], additionally introduced extraction of event anchors, extraction of temporal expressions, and recognition of temporal relations between events from the same sentence, where one syntactically dominates the other. The i2b2 NLP challenge for clinical records [24] uses the same tasks of TempEval-2, which focused on the temporal relations within clinical narratives.

The observations of the present systems, events and times were designed to use for a particular domain (financial, medical). Semantic representation of time is missing in temporal extraction methods.

3 Workflow of the Proposed Method

This section presents a workflow of proposed framework building event-time graphs. The text or the documents are considered to be the input of the work, the output of the system is considered to be the event time graphs.

In the first step text document is fed as input then processed through basic NLP pre-processing steps such as Tokenization, lemmatization, stemming, parts of speech tagging, parsing and Named entity recognition. Thus processed document will be given for extraction of events and times. After the extraction of events and times, the relation between events and times need to be identified for the construction of the event-time graph. The framework proposed is shown in figure-1.

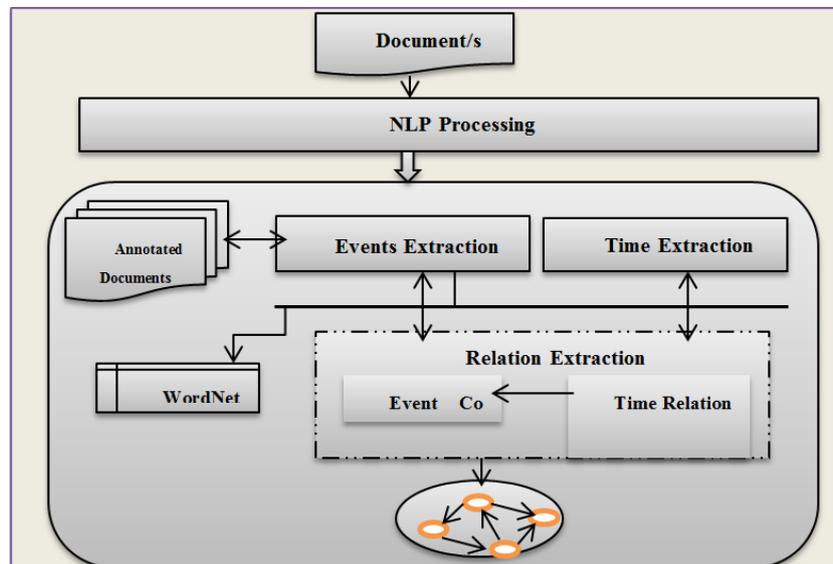


Fig. 1. Workflow of the Proposed Method

3.1 NLP Pre-Processing

Basic NLP pre-processing involves the following steps:

Abbreviation expansion: If the text involves any contractions need to expand, generally these are shortened version of words or syllables. Most of the contractions are exist in English text in written or spoken forms. If English contractions exist in the text that word can be expanded by adding or removing one of the vowels of that word. For example, don't to "do not" and I'd to "I would". Converting each contraction to its expanded, original form helps for text standardization.

Special characters removal: These are symbols or specific characters basically non-alphanumeric characters or even numeric characters (depending on the problem), by using regular expressions (regexes) special symbols can be eliminated from the text.

Tokenization: The process tokenizing means paragraph splitting into sentences and sentences into words or individual words or punctuation through a similar process. Most commonly this split across through the white spaces.

Stop words elimination: Stop words can be articles, conjunctions, prepositions namely a, an, the, etc. These do not form any significance so they are removed from the text.

Lemmatization: Lemmatization removes word affixes to get the base form of a word. The root word is known as the Base form but not the root stem. The difference is that the root word is always a lexically correct word (present in the dictionary), but the root stem may not be so. Thus, root word, also known as the lemma, will always be in the language dictionary.

Parts of speech tagging (POS): POS tagging can be used to tag each word in the text with its POS tag. These tags help to detect the syntax features of the tokens.

For example, consider the following text as input to the proposed framework.

Original Text: Nah I don't think he goes to USA, he lives around here though.

After performing each of the NLP-preprocessing steps the intermediate outputs at each stage are shown below. NLP preprocessing can be with the help as stated in [29].

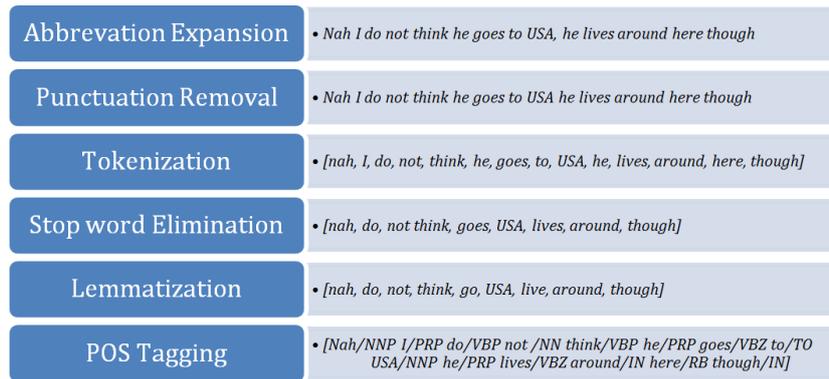


Fig. 2. Steps of stages in Preprocessing

3.2 Event extraction

After the basic NLP pre-processing, the POS tagged tokens from the text will be the input for the event and time extraction component. In English text, events are derived using syntactic and semantic features. An approach is built where the nouns and verbs are extracted. At the initial stage, all the verbs are treated as events. In the next stage, the words that have the noun and verb tags are resolved from ambiguity to identify the nonverbal events.

Given a POS-tagged text as input a triplet is constructed (T, V_t, N_t) where T is the token itself which occurs in the text, V_t is the verb form of the token and N_t is the noun form of the token. Events are referred to be the action forming verbs. In some cases, even the noun forms the events but in this rule based systems it is difficult to

resolve the ambiguous cases of non-verbal events, like for example word “park” can be identified as a noun by the POS tag but it can be a verb also. To identify these cases machine learning approaches are used to predict the semantic role of an event. Machine learning approaches like Conditional Random Field (CRF), Semantic Role Label (SRL) with WordNet are used to resolve nonverbal events.

Conditional Random Field (CRF) is one of the discriminative classifiers which correspond to conditionally trained the probabilistic model. The conditional probability of a state sequence $X = \{x_1, x_2, \dots, x_T\}$ given an observation sequence $Y = \{y_1, y_2, \dots, y_T\}$ function $f_k(x_{t-1}, x_t, y, t)$ is a feature function whose weight k is to be learned via training. CRF considers the tense and aspect of a lexicon for event detection. The main advantage of using CRF is that it has sequential data handling after recognition of one token the previous context of that token can be taken into consideration to calculate the conditional probability. But still, some nonverbal lexicons were not resolved as events.

To overcome the problem Semantic Role Labelling (SRL) WordNet are used to detect nonverbal events. SRL extract all constituents determining their arguments and their adjuncts (locative, temporal) of a lexicon. Nominalization is the process to get the constitutes with the help of its suffixes (-ed, or, ee, er). Event nominalizations often afford the verbs and if nominalizations are defined as nouns, morphologically derived from verbs, usually by suffixation. For example, consider a sentence:

“All sites were inspected to the satisfaction of the inspection team and with the full cooperation of Mumbai authorities, Ram said.”

SRL output for the sentence is as follows:

[ARG1 All sites] were [TARGET inspected] to the satisfaction of the inspection team and with the full cooperation of Mumbai authorities, [said].

In the first traverse, the word “inspected” identified as the event. In the second traverse, “said” is identified as an event. All the extracted target words are treated as event words. It is observed that many of these target words are identified as the event expressions but there exist many nominalized event expressions (i.e., deverbal nouns) that are not identified as events by the supervised CRF. These nominalized expressions are correctly identified as events by SRL. Even after identifying semantic roles still, there are some cases where noun-verb ambiguity exists, for some of the words like ‘war’, ‘attempt’, ‘tour’. To overcome this WordNet is attached to SRL. The stem of noun words are checked by WordNet if any one of the word senses as a verb then it will be announced as event and if the token appears as noun and verb then also it is treated as an event. (Eg: “Declared” is a word the stem word is “declare”). WordNet features have been widely used to extract different lexical categories, such as part-of-speech (POS), stem, hypernym, meronym, distance, and common parents, and demonstrated its worth in many tasks. Here, WordNet is mainly used to identify non-deverbal event nouns. There is a performance improvement with the integration of WordNet to SRL and CRF methods.

For example, given a text: *Dr. Abdul Kalam is known as the ‘missile man’, born on 15th October 1931 in Tamil Nadu. Kalam graduated in science from St. Joseph’s College, 1954. He reports every day in lab his aim is vision space.*

The output of the NLP preprocessing step is *Dr/NNP Abdul/NNP Kalam/NNP known/NN as/IN the/DT â/NNP €/NNP missile/NN manâ/-NONE- €™,/. born/NN on/IN 15th/CD October/NNP 1931/CD in/IN Rameswaram/NNP ./, Tamil/NNP Nadu/NNP. ./, Kalam/NNP graduated/VBD in/IN science/NN from/IN St/NNP ./.* *Joseph/NNP 'POS s/NNS College/NNP/1954/CD. Kalam / NNP reports/NN in/IN lab/NN aim/NN vision/NN space/NNP.*

The output of the POS tagger: *Dr/NNP Abdul/NNP Kalam/NNP known/NN missile/NN manâ/-NONE- €™,/. born/NN Rameswaram/NNP Tamil/NNP Nadu/NNP graduated/VBD science/NN St/NNP Joseph/NNP College/NNP/ reports/NN lab/NN aim/NN vision/NN space/NNP.*

After passing through conditional random field, SRL with wordnet the following words are the output i.e, extracted as events.

“Missile”, “born”, “graduated”, “science”, “report”, “aim”.

The output of the Event Extraction task of the framework extracts the events by using the syntactic and semantic feature from a given natural language text irrespective the domain. Integration of CRF, SRL, wordnet with hand-coded rules addressed the gaps identified in the existing events extraction systems.

3.3 Time extraction

Time in the NLP text exists in quantitative and qualitative forms. Several methods were developed in the past to extract various forms of time expressions. In this framework quantitative time expression i.e. calendric times mentioned with specific date time in standard ISO format were directly recognized by using SuTime. Time expressions like “independence day”, “Mother’s day” are not captured by SuTime. To extract such temporal expressions, pattern-based rules are developed and integrated into the existing framework. Rules are developed for calendric holidays of INDIAN scenario. Below are some set of rules from a holiday package.

*//Sample rules {/new//year/\$POSS?/day/?)=>IsoDate(NIL,1,1)//January1st
{/republic/ /day/)=> IsoDate(NIL, 1, 26)}// RepublicDay
Day {(/independence//day/)=>IsoDate(NIL, 8, 15) //Independence Day*

Extraction of Time Expressions: Quantitative time expressions are in a numeric form where qualitative forms are not. Time expressions are obtained in lexical, syntax and semantic features.

- Quantitative time expression as lexical features these are calendric times mentioned with specific date time in standard ISO format these quantitative time directly recognized by using SuTime.
- Qualitative time expression as a syntactic feature of the time represented with Allen’s algebra with 13 forms of relations (Before, after, during, overlap) between events these are not directly specified with a number. In time expression representation semantic feature of time considers specific holiday events like (independence day, mother’s day, etc..) these semantic representations of time not captured by SUTime, for that we implemented pattern-based rules as added as holiday package to SuTime.

- SuTime doesn't capture Qualitative time 13 relations of Allen's and SuTime does not identify semantic time relations like Indian holidays. In our work by using lexical, syntax features es differentiating a number as a numeric value and a time expression with time relation. By adding holiday package to the SuTime where it will extract time in semantic form like (*Independence Day, Mother's day of Indian calendaric holidays*).
- The overall process of time expression first lexical lookup feature means mapping names to numbers, units to ISO values, etc. Next is context-dependent classification: determining whether the time is a point or duration, looks forward or backward, makes specific or generic reference with the help of lexical, syntax and semantic features. The reference time for time expressions whose values must be computed in final computation combining the results of all of these steps to produce a final normalized value to detect time expression

For Example the given Text: Kalam was born on 15th October 1954. On Independence day discussed kalam's life history. He visited Delhi after 26th January.

The output after the tokenization: *Kalam//NNP was//VBD born//VBN on//PP 15th//JJ October//NNP , 1954//CD. On //IN Independence//NNP day//NN discussed//VBD kalam//NN life //NN history//NN. He//PRP visited//VBD delhi//NN after //Prp 26th//JJ January//NNP*

Output after the time Expression recognition:

*15th October 1954 Independence Day After 26th January/
// "15th October" "1954" "26th January" are Quantitative time, Independence day semantic time After is qualitative time.*

3.4 Relation extraction

In this step, the relationship needs to establish between the extracted events and time expressions. Event reference relation and time relation declares the final events and time expressions with the lexical, syntax and semantic features and that are useful for event-time graph construction from the extraction modules.

The rules to declare the events and time from the given text by the following rules:

- **Suffix Semrule:** In Morphological way nouns which are derived from verbs distinguished as nominalizations (or, deverbal nouns). The deverbal nouns are usually identified by the suffixes like '-tion', '-ion', '-ing' and '-ed' etc. The nouns that are not NEs, but end with these suffixes are considered as the event words.
- **Verb and Noun Semrule:** The verb-noun combinations are searched in the sentences of the test set. The non-NE noun word tokens are considered as events.
- **Nominal and nonverbal Semrule:** Nominals and non-deverbal event nouns can be identified by the complements of aspectual PPs headed by prepositions like during, after and before, and complex prepositions such as at the end of and at the

beginning of, etc. The next token(s) appearing after these clue word(s)/phrase(s) are considered as events.

- **Event Noun Semrule:** The non-NE nouns occurring after the expressions such as frequency of, the occurrence of any period of are most probably the event nouns.
- **Object Semrule:** Event nouns can also appear as objects of aspectual and time-related verbs, such as “*have begun a campaign*” or “*have carried out a campaign*”. The notion of Named Entity is that appear after the expressions like “have begun a”, “have carried out an” etc. are also mostly events.
- **Mapping of the rules to declare the time events by the following rules:** The events and time expressions are obtained from the above individual steps next is to the map the events which associate with time expressions that are said to be temporal expressions or time expression. In these events which relates to other events can be noted with qualitative time expressions, that will be the relations between events.

4 Event – Graph

A graph is the representation of events and time relations between events or concepts. In our work, the concepts of individual real-world events those are present in the text as an event. We define an event graph as a vertex- and edge-labeled mixed graph in which vertices represent the individual events and edges represent the temporal reference relation between events. Given that some relations between events are symmetric (temporal overlap, before, after, temporally equal), an event graph may contain both directed and undirected edges; thus, we define an event graph as a mixed graph.

- Let an event-time graph G in tuple notation $G = (V, E, S, m, r)$, ‘e’ is set of extracted events, ‘r’ denotes the set of relations. In tuple notation G , where V is the set of vertices, E is the set of undirected edges, S is the set of directed edges (arcs), $m : V \rightarrow e$ is a vertex labeling function mapping vertices to event from ‘e’ and $r : E \cup S \rightarrow r$ is an edge labeling function mapping edges to relations from r . Each event from ‘e’ is assigned a type (e.g., Occurrence, Reporting) and consists of event features and a set of typed event arguments.
- In our work, we define the set of relations with events ‘e’ and relations ‘r’ based on the following:
 - **Fact feature of the events:** in real-world events represents facts, Eg: in question answering for factual event is (e.g., ‘Who killed MahatmaGandhi?’), and non-factual event (e.g., ‘Who did not win a gold medal in 2016 Olympics?’ or ‘When might Clinton resign?’). In our work we didn’t focus nonfactual events or hypothetical (e.g., ‘She can win’), future (e.g., ‘She will win’), negative (e.g., ‘She did not win’), and counterfactual event mentions (‘If he had won ’), Since we aim to represent factual event are real-world events which actually occurred, also essential for the temporal ordering of events (Karttunen and Zaenen 2005) because non-factual events can hardly be placed on a timeline.

- **Event features** consisting of:
 - **Token features:** word, lemma, stem, ending and POS tag.
 - **Context features:** These are all of the token features (e.g., word, lemma, stem) computed for all the tokens in a two-token symmetric window.
 - **Syntactic features:** These include all syntactic dependency relations of a token, the chunk types (e.g., NP, VP, PP) of the token chunk and its adjacent chunks.
 - **Modifier features:** These aim to discriminate between factual and non-factual events by describing different modifiers for the token.
 - **Other features:** These include a feature indicating whether the word is capitalized (named events, such as ‘World Cup’ are capitalized).
- **Event argument types:** In extraction to get the robustness, we considered four types of event arguments which are an agent, target, location, and time. These types of arguments are suitable to answer four main wh-questions: ‘where, and when, who did what to whom?’
- **Types in relations:** Many types of semantic relations may exist between events (e.g., temporal, causal, semantic relations). In our work, we focus on two types of relations temporal relations (because events are temporally determined concepts; Pustejovsky et al. 2003b) and event reference relations (because a single real-world event may be referred to by multiple relations).

In a document event references, we do not merge the vertices, each graph vertex represents a single event mention and not a real-world event referred to by the event references.

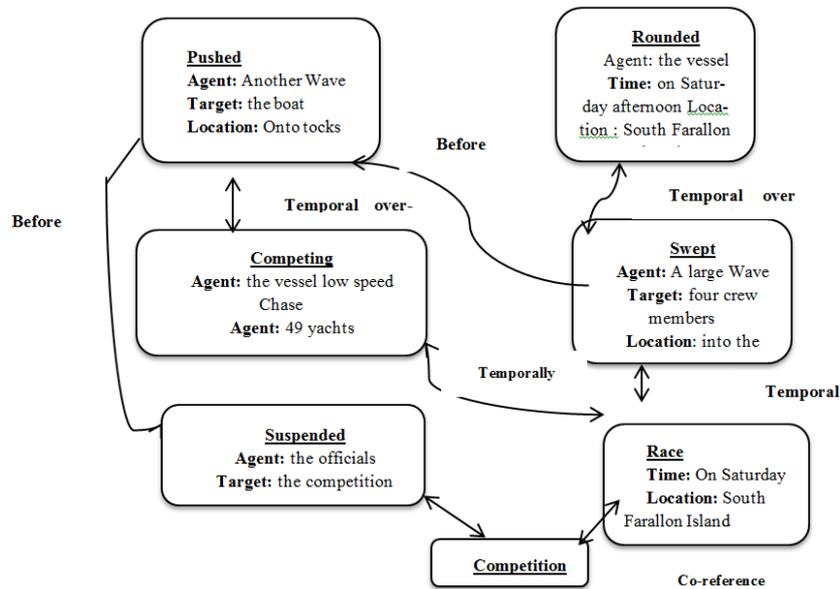


Fig. 3. Example Reference event graph from a news story from news corpus

Input text for the above graph: “The boat was pushed on the rocks. A wave swept four crew members Stallion rounded Farallon Island Stallion allowed Chase a head start Stallion raced on Sunday, Stallion raced around Farallon Island around Stallion competed against Chase. The boat sunk on the shore ‘The boat which won the race. The crewmen storm swept’ The first boat to beat Stallion was Chase Yesterday’s race ‘Stallion’s victory ’ ‘Farallon Island race’(x = ‘race’, y = ‘Island’) ‘Sunday competition’ Stallion managed to get past Chase”.

5 About the Data Set or Corpus

In our work we used the corpus of Forum for Information Retrieval (FIRE-2018) Datasets contains various semi-structured documents (XML) consisting of events related to 11 categories of files. The categories emphasized are Accidents, Crime, Cyclone, Earthquake, Fire, Floods, Shootout, Storm, Suicide Attack, and Volcano.

By nature, the document is in English language and not domain specific, After training the Model with FIRE [27] dataset we faced an issue, with fewer data. This less data isn’t helping to train the model accurately. Thus we started scraping news articles from news wires and social media. We created a News data set which falls under the same 11 categories.

// Sample document from the newly formed dataset:

“A 23-year-old man was taken to Rockhampton Hospital after being trapped inside his car. His injuries were not life-threatening. The 52-year-old driver of the vehicle and two other people were not injured. Traffic diversions have been put in place until the scene is cleared. A seven-year-old boy was taken to Theodore Hospital for spinal precautions after a two-vehicle crash near Dingo earlier this morning.”

The total we created 200 documents of news because news data contains more event information. In this, each document annotations are removed and 160 used for training and 40 documents for the test set. In that 160 documents also through cross-validation considered for train and test in 10 cross folds, and measures are precision-recall f-measure the accuracy applied for the various classifiers, the results section presents the values of each step.

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. Thus the final text in the document will be plain text without any XML tags

6 Evaluation Outputs and Discussions

6.1 Evaluation of event extraction

The data set used for evaluation of proposed work by using SemEval-2010 as stated in [28] from TempVal and MUC to get the evaluation results. SemEval and MUC datasets are normal text documents but not specific to any domain. First hand-coded

rules were applied on preprocessed input, next CRF based system along with WordNet was applied. Hand-coded rules are developed by using lexical information of the word, CRF technique is implemented with various features of the word, CRF and WordNet combined to extract more nonverbal words.

SemEval dataset as stated in [30] contain 323 verbal and 125 non-verbal event nouns. Table-1 presents the results of the proposed framework. Evaluation metrics such as e precision, recall, and F-measure were used to gauge the performance of the proposed work. Performance of F-Measure increases by at least 1.84% from base hand-coded rules to the hybrid method. Table-1 shows very high-performance improvement (i.e., 3.98%) with the use of CRF+wordNet. The Inclusion of the hybrid/composite method rules obtained an increased F-measure of 6.43% when compared with earlier methods. The precision, recall, and F-measure were calculated for the proposed hybrid model for event extraction and the values were found to be 81.11%, 84.23% and 82.64%, respectively. An improvement of approximately 6% F-measure is obtained when compared with other base methods.

MUC dataset [30] contain 200 sentences over the 30 documents and 253 verbal and 110 non-deverbal event nouns, in Table-2 results were presented. Performance of F-Measure increases by at least 2.54% from base hand-coded rules to the hybrid method. Table-2 shows very high-performance improvement (i.e., 3.99%) with the use of CRF+wordNet. The Inclusion of the hybrid/composite method rules obtained an increased F-measure of 7% when compared with earlier methods. The precision, recall, and F-measure were calculated for the proposed hybrid model for event extraction and the values were found to be 73.38%,72.21% and 73.74%, respectively. An improvement of approximately 7% F-measure is obtained when compared with other base methods.

Table 1. Results for Event Extractin MUC DataExample table

Method	Precision	Recall	F-measure
Handcoded Rules	75.32	77.14	76.21
CRF	77.21	78.91	78.05
CRF+ WordNet	79.20	81.22	80.19
Our Event Extraction Model	81.11	84.23	82.64

Table 2. Results for Event Extractin MUC DataExample table

Method	Precision	Recall	F-measure
Handcoded Rules	63.32	65.15	64.18
CRF	67.21	68.12	67.64
CRF+ WordNet	70.20	72.21	68.18
Hybrid (or) CompositeRule	72.21	73.38	73.74

6.2 Evaluation of Times Extraction

To obtain the results for the time's extraction form data set for our experiment was collected from Wikipedia articles. We selected the three representative categories of articles as stated in Warfare [26], and Celebrities [26], and news data[26]. Summing

over all articles yields a total of 2000 sentences. We randomly sampled 30 documents with 1200 sentences, within this test set there are total 268 events identified by Evita and total times are 148 hands coded manually the extracted times with the methods presented in table-.

Table 3. Results obtained for the Methods

Test sources/ Articles	No.of Events	Times Retrieved from the Input Data		
		Using SUTIME	Using our Pattern rules Algorithm	SUTIME + Holiday Pack- age of our Algorithm
Warfare	102	28	30	32
Celebrities	78	15	39	32
News data	88	39	69	61

By using the above designations Precision, Recall and F-measure are calculated these measures are the quality measures to find the relevance and accuracy of the methods. In the table-4 total number of times present in the given input are 148, the second row consists of retrieved times after executing the methods, the third row consists of relevant items from the retrieved, and last three rows are accuracy measures.

Table 4. Accuracy of the Results (with Precision, Recall, F-measure)

	Using SUTIME	SUTIME+ Holiday Package of our Algorithm	Using our Pattern rules Algorithm
Total number of times in the given input	148	148	148
Total Retrieved times	100	125	138
Number of Relevant times from re- trieved	63	118	122
Precision	63%	94%	89%
Recall	43%	79%	82%
F-measure	51%	85.8%	85.3%

Table-4 Shows obtained results for the comparing the three method’s accuracies, using SuTime with holiday package obtained better precision that is 94% and 85% f-measure and our proposed algorithm for times is achieved equivalent f-measure with SuTime with holiday package results. With the above comparisons, the results of our approach also obtained noticeable precision.

6.3 Testing for accuracy with classifiers

In the testing process, 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. To measure the accuracy of the model uses metrics are 5fold cross-validation.

Table 5. Comparison of accuracy parameters with different classifiers

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

	class	important_dates	predicted_class	x_test
12	EARTHQUAKE	[(today evening hour, TIME), (yesterday, DATE)...	EARTHQUAKE	moderate intensity earthquake hit northwest pa...
80	ACCIDENTS	[(sunday, DATE), (morning, TIME)]	ACCIDENTS	van fuel tanker collision kill western afghani...
33	EARTHQUAKE	[(morning, TIME)]	EARTHQUAKE	magnitude earthquake jolt assam kokrajhar eart...
5	SUICIDE_ATTACK	[(morning, TIME)]	SUICIDE_ATTACK	suicide car bomber kill two police official af...
187	FLOODS	[(life past hour, TIME), (month second, TIME)]	FLOODS	least four people die assam last hour state st...

Fig. 4. Snapshot of the output table after testing is done

- Class Column in the table is the actual class of the event (Events).
- The predicted class is the Class which is predicted by our model.
- Important_dates is the column which shows all the important date or time components in the document.
- x_test is our document which is fed to the model.

Accuracy Scores on 5-fold Cross-Validation

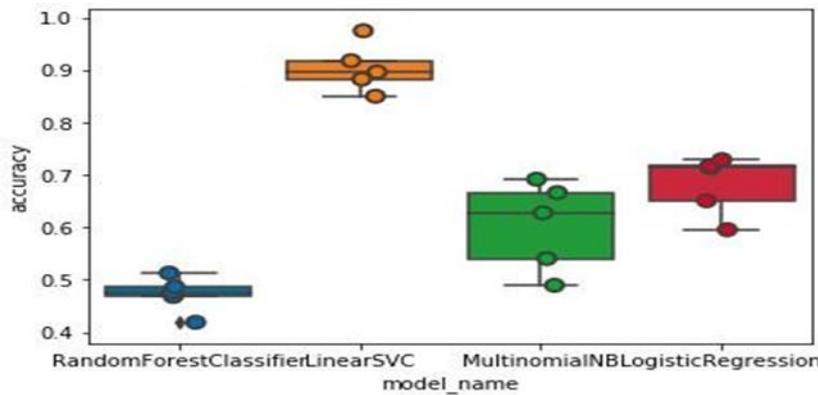


Fig. 5. Accuracy plots of models

Description:

- Each round circle shown in the model is a split of the dataset used as shown in Fig-
- Here 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 6. Calculation of Precision and Recall for Linear SVC

	class1	class2	class3	class4	class5	class6	class7	class8	class9	class10	class11
No of TP	2	3	3	10	2	3	5	23	3	4	2
Total Predicted Positive	2	4	3	10	3	4	7	26	3	4	2
Total Actual Positive	5	4	5	10	4	4	5	25	5	4	3

6.4 Performance of Various classifiers

Random Forests perform slightly worse in the following cases:

- When the dimensionality (number of features) is very high with respect to the number of training samples.
- They fail in sharp corners and exactness. They use diffusion methods. They fit lumpy things well. They do not fit elaborate and highly detailed things well when the sample size is low.

Logistic regression and multinomial naive bayes

- These models are traditional models and they perform modestly better over Random Forests in NLP but very much far away from SVC's

Support Vector Machines: SVM outperforms all the other models by a large margin mainly because:

- With SVM regularization parameter, it avoids the problem of overfitting.
- Convex optimization problems resolved.
- Lastly, it is an approximation to abound on the test error rate, and there is a substantial body of theory behind it which suggests it should be a good idea. That is how support vector machines gave us the best accuracy model.

7 Conclusion

We have successfully identified and classified events with 90% accuracy with effective pre-processing techniques and model. More amounts of data, the accuracy can be further improved. With 80 file got an accuracy of 85%. With 160 files we got an accuracy of 90%. The Linear SVC model performed better than all other models. In the observations of the work is the fact that a better collection of data helped us in training the model better.

7.1 Future scope

Further if the data is more it may improve the accuracy. With more data, deep learning models which work very well on huge amounts of data. And this work can also make this as a web service which takes the article as an input and gives the event class and important dates of the article and identifies event time relationships. This model can be deployed in any real-time application to help readers to get articles based on the categories.

8 References

- [1] Allan, J. 2002. Topic Detection and Tracking: Event-Based Information Organization, vol. 12.
- [2] Salton, G., Wong, A., and Yang, C. 1975. A vector space model for automatic indexing. *Communications of the ACM* 18(11): 613–20. <https://doi.org/10.1145/361219.361220>
- [3] Hatzivassiloglou, V., Gravano, L., and Maganti, A. 2000. An investigation of linguistic features and clustering algorithms for topical document clustering. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY: ACM, pp. 224–31. <https://doi.org/10.1145/345508.345582>
- [4] Makkonen, J., Ahonen-Myka, H., and Salmenkivi, M. 2004. Simple semantics in topic detection and tracking. *Information Retrieval* 7(3): 347–68. <https://doi.org/10.1023/b:inrt.0000011210.12953.86>
- [5] Ahn, D., Schockaert, S., De Cock, M., and Kerre, E. 2006. Supporting temporal question answering: strategies for online data collection. In *Proceedings of the 5th International Workshop on Inference in Computational Semantics*. Buxton, UK: ACL, pp. 127–32.
- [6] ALLAN, J., R. PARKA, and LAVRENKO V. 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, SIGIR '98*, ACM, New York, NY, pp. 37–45. <https://doi.org/10.1145/290941.290954>
- [7] Grishman, R., and Sundheim, B. 1996. Message understanding conference-6: a brief history. In *Proceedings of International Conference on Computational Linguistics (COLING 1996)*, Copenhagen, Denmark, vol. 96. Dresden, Germany: ICCL, pp. 466– 71. <https://doi.org/10.3115/992628.992709>
- [8] Humphreys, K., Gaizauskas, R., Azzam, S., Huyck, C., Mitchell, B., Cunningham, H., and Wilks, Y. 1998. University of She eld: description of the LaSIE-II system as used for MUC-7. In *Proceedings of the Seventh Message Understanding Conferences (MUC-7)*, San Diego, CA. Gaithersburg, MD: NIST. <https://doi.org/10.3115/1072399.1072418>
- [9] Aone, C., and Ramos-Santacruz, M. 2000. REES: a large-scale relation and event extraction system. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*. Seattle, WA. Stroudsburg, PA: ACL, pp. 76–83. <https://doi.org/10.3115/974147.974158>
- [10] Yangarber, R., Grishman, R., Tapanainen, P., and Huttunen, S. 2000. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th Conference on Computational Linguistics, Hong Kong*, vol. 2. Stroudsburg, PA: ACL, pp. 940–6. <https://doi.org/10.3115/992730.992782>

- [11] Zhen Lei, Ying Zhang, Yu-chi Liu, et al. A system for detecting and tracking internet news event. In *Advances in Multimedia Information Processing-PCM 2005*, pages 754-764. Springer, 2005. https://doi.org/10.1007/11581772_66
- [12] S. Bethard and J. H. Martin, "Identification of event mentions and their semantic class," presented at the *Empirical Methods in Natural Language Processing (EMNLP)*, 2006. <https://doi.org/10.3115/1610075.1610098>
- [13] R. Sauri, et al., "Evita: a robust event recognizer for QA systems," presented at the *Human Language Technology and Empirical Methods in Natural Language Processing*, 2005 Sauri et al., 2005), Piskorski et al. (2007), REES (Aone and Ramos-Santacruz, 2000), Nishihara et al. (2009), Hung et al. (2010) <https://doi.org/10.3115/1220575.1220663>
- [14] Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the COLING-ACL, Montreal*. FrameNet (Baker et al. 1998) or VerbNet (Kipper et al. 2000), BioNLP (Jayalakshmi et al. 2014). <https://doi.org/10.3115/980845.980860>
- [15] Chinchor, Nancy. 1999. Overview of MUC-7/MET-2. In *Proc. Message Understanding Conference MUC-7*.
- [16] Kirk Roberts, Bryan Rink, and Sanda M Harabagiu. A exible framework for recognizing events, temporal expressions, and temporal relations in clinical text. *Journal of the American Medical Informatics Association*, 20(5):867-875, 2013. TempEx Mani and Wilson, 2000. <https://doi.org/10.1136/amiajnl-2013-001619>
- [17] Angel X Chang and Christopher D Manning. SUTime: A library for recognizing and normalizing time expressions. In *LREC*, pages 3735-3740, 2012. SUTime Chang and Manning, 2012
- [18] Jannik Strotgen and Michael Gertz. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321-324. Association for Computational Linguistics, 2010. Heideltime, Strotgen, and Gertz, 2010. <https://doi.org/10.3115/v1/e14-4026>
- [19] Pustejovsky, James, Jos'e M. Casta~no, Robert Ingria, Roser Sauri, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003a. TimeML: Robust Specification of Event and Temporal Expressions in Text. In *IWCS-5*.
- [20] J. Pustejovsky, et al., "TimeML: Robust Specification of Event and Temporal Expressions in Text." in *New Directions in Question Answering*, 2003. J. Pustejovsky, et al., "The TIMEBANK corpus," 2003. Pustejovsky et al. 2003b.
- [21] Mani, I., B. Wellner, M. Verhagen, and J. Pustejovsky. 2007. Three Approaches to Learning TLINKs in TimeML. Technical Report CS-07-268, Computer Science Department, Brandeis University, Waltham, USA.
- [22] ACE. 2005. Evaluation of the Detection and Recognition of ACE: Entities, Values, Temporal Expressions, Relations, and Events. Gaithersburg, MD: NIST.
- [23] Verhagen, M.; Gaizauskas, R.; Schilder, F.; Hepple, M.; Katz, G.; and Pustejovsky, J. 2007. Semeval-2007 task 15: TempEval temporal relation identification. In *SemEval-2007: 4th International Workshop on Semantic Evaluations* Verhagen et al. 2007). <https://doi.org/10.3115/1621474.1621488>
- [24] Sun W, Rumshisky A, Uzuner O. Evaluating temporal relations in clinical text: 2012 i2b2 Challenge. *J Am Med Inform Assoc*. 2013;20(5):806-813. doi:10.1136/amiajnl-2013-001628. <https://doi.org/10.1136/amiajnl-2013-001628>
- [25] TRIPS and TRIOS System for TempEval-2: Extracting Temporal Information from Text Naushad UzZaman *Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010*, pages 276-283.
- [26] en.wikipedia.org/wiki/Warfare: Warfare, en.wikipedia.org/wiki/Celebrity, and News data.

- [27] “Event Extraction using support vector machines” - FIRE 2017.
- [28] Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G., Pustejovsky, and J.: SemEval-2007 Task 15: TempEval Temporal Relation Identification. Proceedings of the 4th International Workshop on Semantic Evaluations (semEval-2007), pp. 75-80, Prague. <https://doi.org/10.3115/1621474.1621488>
- [29] <http://nlp.stanford.edu/software/lex-parser.shtml>
- [30] http://semeval2.fbk.eu/semeval2.php?location=download&task_id=5&datatype=trial
- [31] Strotgen, J., Gertz, M.: Heildeltime: high quality rule-based extraction and normalization of
- [32] temporal expressions. In Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 321–324. Association for Computational Linguistics (2010)

9 Authors

Vanitha Guda is 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. S Suresh Kumar is presently working as Assistant professor and Head in the Department of Information Technology at JNTUH College of Engineering Jagtial. He has PHD in Natural Language processing from JNTUH. His area of specialization includes Natural Language Processing, Information retrieval systems, etc. He published nearly 20 publications in various journals and conferences.

Article submitted 2019-06-03. Resubmitted 2019-07-25. Final acceptance 2019-07-27. Final version published as submitted by the authors.