Building Graph for Events and Time in Natural Language Text

Vanitha Guda , SureshKumar Sanampudi

Abstract*: *Events and time are two major key terms in natural language processing due to the various event-oriented tasks these are become an essential terms in information extraction. In natural language processing and information extraction or retrieval event and time leads to several applications like text summaries, documents summaries, and question answering systems. In this paper, we present events-time graph as a new way of construction for event-time based information from text. In this event-time graph nodes are events, whereas edges represent the temporal and co-reference relations between events. In many of the previous researches of natural language processing mainly individually focused on extraction tasks and in domain-specific way but in this work we present extraction and representation of the relationship between events- time by representing with event time graph construction. Our overall system construction is in three-step process that performs event extraction, time extraction, and representing relation extraction. Each step is at a performance level comparable with the state of the art. We present Event extraction on MUC data corpus annotated with events mentions on which we train and evaluate our model. Next, we present time extraction the model of times tested for several news articles from Wikipedia corpus. Next is to represent event time relation by representation by next constructing event time graphs. Finally, we evaluate the overall quality of event graphs with the evaluation metrics and conclude the observations of the entire work.

Keywords: Events, Time, Event-Time Graph, Question Answering systems.

I. INTRODUCTION

In real world context events are nothing but happenings, situations and an important building term to explain situation. Events performs major role while doing document understanding such as text summaries, news-summarization (VossenandCaselli,2015), information extraction (Chambers and Jurafsky, 2011) and story understanding (Mostafazadeh et al.,2016). Another definition for event in textual form events can also called as linguistic events, or event mentions, and these are preferred in computational linguistics (Rosen 1999).

Revised Manuscript Received on January 30, 2020.
***** Correspondence Author

Vanitha Guda*, Department of CSE,Chaithanya Bharathi Institute of Technology(A),Hyderabad, India.

Suresh Kumar Sanampudi, Department of IT JNTUCEJ, Nachupally Jagityal Karimnagar,India

In generic way events may fall within some time stamp or location but in real world two events at the same location and exactly at same time may or may not happen (Quine 1985).

Sometimes event may contain vague in nature because the exact timestamp or location of an event may not always inferred from text. Real-world events mostly contain some temporal relation like after, before etc., in many situations the relationship may not present. Event described with several features like location, and several entities/participants, a time interval. In languages text most of the form of events described as in verbs and nouns that indicates state changes (Vendler, 1957) and explains text spans are nothing but event mentions or nuggets of an event.

In this paper, we present a study in two types of relation: Event Sequencing (ES) and Event Co-reference by hopper (EH). Two events are said to be Co-referent if they refer to the conceptually same underlying event, even if their arguments are not strictly identical the "same" and event may be realized by multiple text spans((Liu et al., 2014).

Event Sequencing (ES), on the basis of event co-reference relations various relations are established between the events. Event sequencing task is nothing but reassembling the events in sequences and expresses them with the relation. Schank's scripts motivated the task of ES (Schank and Abelson, 1977), which explains humans organize information through procedural data structures. Event sequencing explains about how to perform the grouping the events and ordering of the text documents belongs to the same document. By taking a document, ES performs the identification of events within the text and classifies the internal relations. Relations can be represented by Directed Acyclic Graphs (DAGs) labeled ones, there are two types of relations one is relations connect several events with subevents, another is events connects with time constraints. In this paper, we focus on how events connect with other events and events with time relations.

ES task, needs to perform the temporal ordering event sequences for the sequencing it considers temporal ordering means events to lay on timeline, similar time points for the events may be complicated. To handle events co-reference with a time line points a decoding algorithm proposed (Bjorkelund and Kuhn, 2014; Durrett and Klein, 2014; Lee et al., 2017). For structure prediction, decoding of sentence structure from local decisions is one of the major problem. However, unlike co-reference relations, sequencing relations are directed. Co-reference decoding algorithms cannot be directly applied to relations. We present a graph-based representation for event sequencing or occurrence.

The remaining part of the paper is explained like in: the related works on event, time extraction and the structuring of event-information in section 2.

We explain experimental setup and model to construct event graphs with implementation details in section-3. We present results with obtained outputs of each step and evaluated results of the overall process in Seciton-4. Conclusions and future work in section-5.

II. RELATED WORKS

An Event co reference is the main term in many researches the tasks in event co-reference involves (Humphreys et al. 1997). The Recent works mentions event co-reference as the availability of annotation for event tasks. In many ways event co-reference is defined with different definitions and assumptions because of the complexity in existence. In ACE event co-reference researches are conducted on the popular on ACE corpus (A rock, 2012; Sangeetha and Chen et al., 2009; Chen and Ji, 2009 Chen and Ng, 2015, Chen and Ng, 2013). In ACE task event co-reference is restricted for argument matching. Events are majorly assosiated with restricted domain (Hovyet al.,2013;Cybulska and Vossen,2012). Another is to in event extraction considering the issues within-document and also cross-document co-reference (Cybulska and Vossen, 2014 and Lee et al, 2012) works on ECB corpus. Event sequencing is also a major step for event extraction in this supervised method on explained by (Araki et.al, 2014). Another works are (Allen and Chambers et al., 2014) explained about temporal relations. Subsequently many works are done by Frame Net or Prop Bank (Chambers and Jurafsky2009). The proposed temporal awareness metric has already been used for the TempEval-3-shared task (Uzzaman et al. 2013). In model building, the metrics considers the structure for temporal constraints to identify the temporal relations. Event anchors and features the summaries were built-in document with all sentences which are the same features with existing works (J. Mooney. 2016) of Event Sequencing task is motivated on common-sense knowledge about the event chronology. Our work flow is to explain about events, relations between events and extracting and representing events and times. Event sequencing is to find the other temporal relations between events. In Time bank corpus (Pustejovsky et al., 2002) temporal detection is mainly focused on recent computational approaches.

III. EXPERIMENTAL SETUP AND MODEL

Our work model aims to build the model for representing event time relation by extracting and representing event time information from raw text. Model consists of three primary modules one is events extraction, times extraction, relation identification and representing in event time graph. In this work for events extraction not considered any specific domain and time extraction semantic features of time are considered. In this paper we focused mainly to represent event time graphs construction.

In figure-1 it is represented with modules of event extraction and time extraction and time related events are the major modules. After the extraction of the modules time related events identification is the separate module and event time graphs are represented in a separate module for the construction we considered few relations like event to event with time and event with time point, generic vague nature events are ignored in our work because that doesn't effects the final extraction of time related queries.

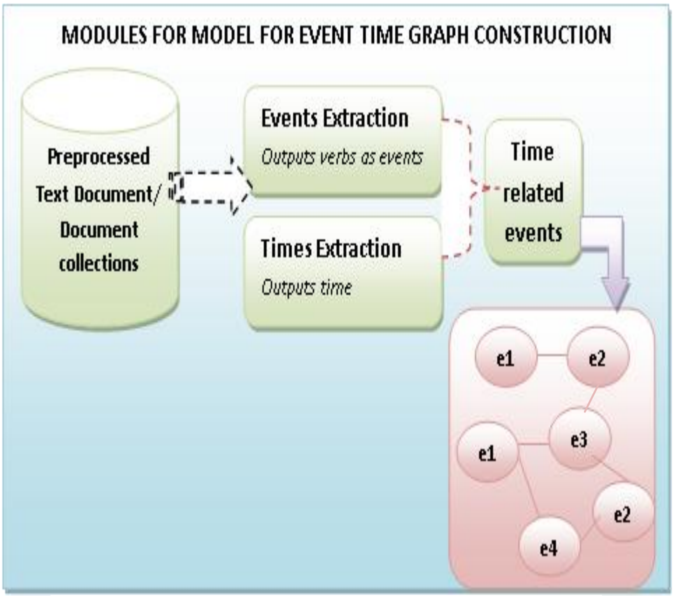

Figure 1: Modules to construct event time graph

In figure-1 the complete model with modules to build event time graph, the components are individual in its functionality with state of the art. For the entire model preprocessed single document or set of documents are the input for the model for the modules of events extraction and time extraction modules. In section 3.1 modules of the 3.2 experimental setups and implementation of modules explained in detail.

3.1 Major Modules of the Model

i) Event Extraction: In Event extraction considers the pre-processed text in the form of token which are tagged with parts of speech by POS tagger. Syntactically identified tokens are the input for the event extraction and time extraction modules. Basically events in literature form are represented with syntax and semantic features. In first iteration all the verbs are treated as events so basic verbs are considered as events. In next step identified some words are mentioned as noun and verb that are ambiguous in nature treated as nonverbal events. By using several machine learning approaches like CRFs and SRL and WordNet with several framed rules events were extracted.

ii) Times Extraction: In general way time represented as quantitative and qualitative. Quantitative are calendric times or specific date and time in standard ISO format and another form is qualitative forms which are represented with relations or duration after, before two days ago etc. Quantitative time expression can easily detected by SuTime or based on the document creation time. But some specific form of time expressions like "Mother's Day", "independence day", and the events related to specific type of time that events are not recognized by SuTime. In this work to identify and extract such time expressions we have built several pattern-based rules which are added to existing framework.

iii) Time related Events: In this module after identification of events and times considering the events which are related to time expression that are treated as temporal events, the events with time point. To build events and time related graphs this step most important with extracted events and time points.

All other events which are not connected with any kind of time point are non-temporal events.

iv) Events and time Graphs: Event-Time graph structure constructs events as nodes and labeled edges represent relations. vertices are events, edges are relations between events and time expression. Graph formal notation represented as G and with notations are V for vertices, E for edges, m is mapping and r is the relation that can be finally stated as $G = (v, e, m, r)$, these notations are used in graph representation. In experimental setup 3.2 explains example for event-time graph.

3.2 Experimental setup and Implementation details

Step 1: **Preprocessing**

Preprocessing is the basic step for any natural language process, text need to be aligned in a structured form. Basically preprocessing stage text will be treated as in the form of tokens, the required tokens will be processed in lexical and syntactic form.

- Preprocessing of text in python using predefined major functions are ***import nltk*** which contains major libraries for NLP, for string operations like removing the punctuations and lower upper case characters can be handled by importing the ***import string*** library.
- Stop words which doesn't effects the extraction of the meaning full token, the stop words in English text can be handled by using the package ***from nltk.corpus import stopwords.***
- Proper stem the root word of the token can be handled by using ***from nltk.stem.porter import PorterStemmer().***
- Tokenization is the primary task in preprocessing this can be handled by several tools in python nltk, textblob, spacy. In our work we used nltk the packages can ***from nltk.tokenize import word_tokenize stemmer=Porter Stemmer (from nltk.tokenize import word_tokenize).***
- POS Tagging : Nltk parts of speech tagger tags the tokens with syntax ***import pos tagger from nltk import word brown tagger*** (pos tagger).

Step1 performs preprocessing and generates tokens which are syntactically tagged by POSTagger. Preprocessed tokens are the inputs to events extraction and times extraction modules.

Step2: **Implementation of Events Extraction**

In English linguistics events exists in the form of verbs. In preprocessing the syntactically tagged tokens clearly tagged by parts of speech with syntax from. Extracting verbs from the syntactic tagged tokens are simplest but the main issue will raise with nonverbal events means some of the token may be acting as noun it may be the verb also example the words like "Park" it may be noun and this can be a verb also it may be treated as events. Identifying the nonverbal is a major issue in order to do this we need to capture all senses of a word.

- To get the non-verbal events in order to get the all senses of a word with the help of the Wordnet and in order to get the sequential word notations with all verb and non-verb words are captured by using conditional random field (CRF).
- By using WordNet, CRF and syntactic rules are formed rules which can recognize and extract events from the given text.

Step3: **Implementation of Times Extraction**

In general way times are Quantitative means numeric form or Qualitative means textual form of representation. Using ISO calendric format all the time notations can be extracted using Regexp or Java SUTIME. But non calendric means another format type notations like "Independents day", "Republic day" these type of textual form of time notations are not identified by SUTIME or TimeMl Timex3 tags. In our work we used SUTIME and Java Regx to recognize lexical and syntactic time expressions, to recognize semantic events we have written pattern based rules for all Indian context time expressions.

***Step 4*:** **Implementation of Time Related Events**

In this step events which are related with quantitative or qualitative time that are called as temporal events. Only these temporal events can provide accurate responses to time based queries in temporal Question answering. In this step other on verbal events are treated as normal events but in order to represent event and time relations we need to consider only temporal events that are identified with time stamps are relations.

Step5: **Event time Graphs**

Graphs are used to model and represent the event time relations by building the event time graph. Events are represented as nodes whereas each node can be an individual point or it may connect with other nodes with relations. Nodes are real-world events mentioned with event attributes. Event-Time graph is edge-labeled graph where edges represent the temporal relations between events attributes.

Rules are formed to extract relations and also avoid the step of disambiguation because the attributes of event argument types are not identified by using syntactic nature. Consider a simple example with different typed arguments like concept, location, and time or cause these all are syntactic way these are prepositional objects and event attributes. In this work major goal is to identify the concept as event, time, or it can be in the form of a relation between event and time. Disambiguation rules are used to avoid ambiguous extraction pattern, these rules are based on the analysis of the development corpus preferring general rules over specific rules that can easily adapt to any other domains.

Here we discuss rules for semantic type of a prepositional object, for semantic types. Order specified for the rules that are listed below:

- Given a token if it is in any form of the verb like VBD, VBN, VB etc,. then it is assigned to event type; or if a given word or token is a named entity of type is verb and noun it can be verified by the events extraction module and treats as events. Complete events extraction algorithm explained in our previous work(VGuda,Suresh 2018) by using that events extracted from natural language text.
- If the given token is cardinal type identification by POS tagger then is a number or time then by the module of time expression will verifies the given is a number or time.

The detailed time expression extraction explained in pattern based rules of our earlier work (VGuda,Suresh, 2017).

- In the given text the prepositional object is represented by time expression then the prepositions are like 'before', 'after', 'during', then the token is considered to be of temporal type.
- According (Derczynski and Gaizauskas 2013), temporal signals are such as 'before' and near – are ambiguous and can have both temporal and location as spatial point.
  If it is stated by a temporal preposition like, 'before' 'after', 'during', then the given argument type is a temporal events of the type.
- Graph G can represent the relation between event and time to build the relations. Relations are based on 13 relations of Allen's interval algebra (Allen J, 1983).

IV. RESULTS AND METRICS EVALUATION OF EVENT TIME GRAPH

In this section we present the results of each step how it will considers the given text and what kind of output will be generated after each step. To present the results we considered some random sample text from google and given as input to our model. But to evaluate the complete work we considered standard data sets like SEMEval and MUC Data sets for event and time extraction and EVExtra for event time graph. The outputs of each individual module after the execution are presented in 4.1 and evaluations of obtained results with standard metrics are presented in section 4.2.

4.1 Output results of each Module:

Step 1: Preprocessing step input the file or text

// If the data is from standard dataset file that will be in xml format then the first step of the preprocessing is removing tags from the given fiel converting xml file into txt input.

Given input text

Vallabhai Jhaverbhai Patel was born on 31 October 1875. He organized the party in 1934. Patel native place was Karamsad. Patel passed his matriculation at the late age of 22.Patels marriage was arranged with Jhaverba. The couple had a daughter, Maniben, in 1904.Patels wife Jhaverba was hospitalized in 1909 in Bombay. At the age of 36, he journeyed to England.

- //Removes Stopwords outputs stopword.txt

Vallabhai jhaverbhai patel born 31 october 1875. He organized party 1934. Patel native place karamsad. Patel passed his matriculation late age 22.patels marriage arranged jhaverba. Couple had daughter, maniben, 1904.patels wife jhaverba hospitalised 1909 bombay. Age 36, he journeyed england.

- // Performs Tagging and data is stored in output.txt.pos file

In this step it generates PoS for each and every word which is in the StopWords.txt file as follows

Output after Step1 Preprocessing:

vallabhai/FW jhaverbhai/FW patel/NN born/VBN 31/CD october/NN 1875/CD ./.he/PRP organized/VBD party/NN 1934/CD ./.patel/NN native/JJ place/NN karamsad/NN ./.patel/NN passed/VBD his/PRP$ matriculation/NN late/JJ age/NN 22/CD./.patels/NNS marriage/NN arranged/VBD jhaverba/NN ./.couple/NN had/VBD daughter/NN ,/, maniben/NN ,/, 1904/CD ./.patels/NNS wife/NN jhaverba/NN hospitalized/VBD 1909/CD bombay/NN ./.age/NN 36/CD ,/, he/PRP journeyed/VBD england/NN ./

Note: The above generated output will be the input to events and times extraction module.

Step2: Output after the execution of Events Extraction Module

In this stage in 1st iteration algorithm extracts both noun and verb as events in the next iteration with the help of CRF and WordNet it clearly represents events and the word base form is a event that can also treated as events.

Output after step 2:

Born organized passed arranged hospitalized journeyed

Step 3: Output after the Time extraction Module

In this stage any kind of numerical tokens are represented as cardinals time extraction module differentiates number and time attribute with the help of SuTime recognition and pattern rules and outputs the time expressions.

Output after step 3:

31st October 1875 1934 1904 1909

Step 4: Output after the Time Related Events Module

In this stage sequential arguments are verified with the help of CRF and extracts the events only related to time and that are treated as temporal events will be the output from this step.

Output after step 4:

Born Organized hospitalized

Step 5: Events Time Graph Generation

In generation of Event-time Graph first the dependencies were measured for the verbs that will shows the root word among the relations with all sequence words the sample dependencies are:

Dependencies for word "Born"

Eg: "Dependencies " "-> born-VBN (root) -> patel-NN (nsubjpass) -> vallabhai-NN (nn) -> jhaverbhai-NN (nn) -> was-VBD (auxpass) -> 31-CD (prep_on) -> october-NNP (tmod) -> 1875-CD (num) "

Dependencies Eg: word "Organized"

"Dependencies""->organized-VBD (root) -> he-PRP (nsubj) -> party-NN (dobj) -> the-DT (det) -> 1934-CD (prep_in)

Events –time Graph

Eg: In above example event "born" uttered first in the given text after that "organized".

In above example event "born" uttered first in the given text after that "organized" appeared n text.

4.2 Results with Standard Metrics

To evaluate event time graph by considering the each individual modules of the events time graph model is evaluated in a sequential way. Standard metrics like precision, recall, and F-score are used to evaluate the entre model working. Measuring the performance of the individual steps to get the quality in generating of events and time graph. Table 1 presented results of our work [18].

Table 1: values of Events Extraction Results

Measures	Event Extraction algorithm	TRIPS PARSER	EVITA with Semantics
No. of Events for Test	161	161	161
Total No. of recognized Events	130	110	100
Relevant identified Events	119	95	80
Precision	91%	86%	80%
Recall	73%	75%	78%

❖Figure 1 shows the event extraction results data set used for the evaluation of the model by using MUC, for the evaluation accuracy parameters like precision and recall are considered. MUC (Message Understanding Conference) datasets are normal text documents but not specific to any domain. Results compared with other event extraction models like Trips and Evita.

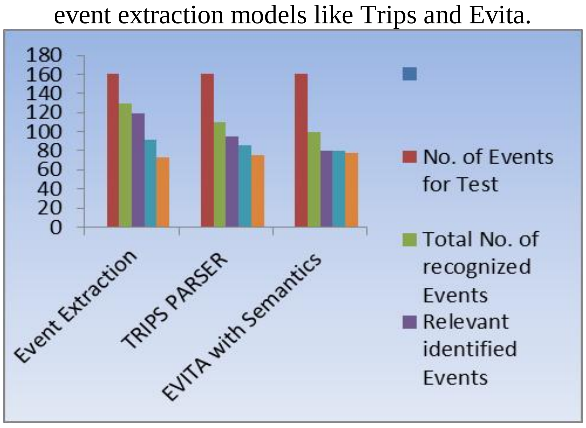

Figure 1: Results of Events Extraction

Table 2: values of Times Extraction Results

	Using SUTime	SUTime+ Holiday Pkg	Pattern rules Algo
Total No.of times in the Input	148	148	148
Total Retrived	100	125	138
no.of relevant Times	63	118	122
Precision	63	94	89
Recall	43	79	82
F-measure	51	85.8	85.3

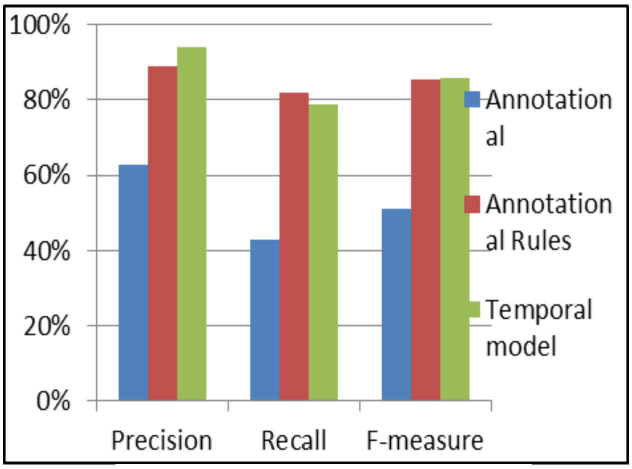

Figure 2: Results of Time Extraction

- ❖ The data set for our experiment was collected from Wikipedia articles of three representative category articles Warfare, Celebrities, and News data. Summing over all articles yields a total 2000 sentences.We randomly sampled 30 documents with 1200 sentences,(840 for training, 360 for test)
- ❖ within this test set there are total 268 events identified and total times are 148. Table-2 obtained results for times extraction of our work [20].

V. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a brief view of the event time graph construction with the terms event and time extraction and representing the relation in natural language text. Overview of the model, extraction methods and implementation steps along with input and output of each step discussed and obtained results got 90% of the precision and reduced recall. Model is independent in domain can easily adaptable to specific domain without disturbing the core modules of the model. Finally Event time graph approaches need high amount of expertise, due to the fact that multiple techniques are combined. Conclusion of the event time relation identification and extraction model performs better in providing responses to temporal questions in question answering systems.

As a future scope of the work this can be extended to applying the model to question answering systems, document summarizations.

REFERENCES

1. Nathanael Chambers and Dan Jurafsky. 2011. Template-Based Information Extraction without the Templates. In HLT '11 Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, pages 976–986.
2. Piek Vossen and Tommaso Caselli. 2015. Storylines for structuring massive streams of news. In Proceedings of the First Workshop on Computing News Story Lines, pages 40–49.
3. Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A Corpus and Evaluation Framework for Deeper Understanding of Commonsense Stories. In Proceedings of NAACL-HLT 2016, pages 839–849.
4. Rosen, S. 1999. The syntactic representation of linguistic events. Glot International 4(2):3–11. Salton, G., Wong, A., and Yang, C. 1975. A vector space model for automatic indexing. Communications of the ACM 18(11): 613–20.
5. Quine, W. 1985. Events and reification. In E. LePore and B. P. McLaughlin (eds.), Actions and Events: Perspectives on the Philosophy of Donald Davidson, Oxford UK: Blackwell, pp. 162–71
6. Vendler, Z. 1957. Verbs and times. The Philosophical Review 66(2): 143–60.
7. Jun Araki. 2018. Extraction of Event Structures from Text. Ph.D. thesis, Carnegie Mellon University.
8. Zhengzhong Liu, Jun Araki, Eduard Hovy, and Teruko Mitamura. 2014. Supervised Within-Document Event Coreference using Information Propagation. In Nicoletta Calzolari (Conference Chair) Khalid Choukri Piperidis, Thierry Declerck, Hrafn Loftsson, Bente Maegaard Stelios, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pages 4539–4544, Reykjavik,
9. Jing Lu and Vincent Ng. 2017. Joint Learning for Event Coreference Resolution. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 90–101.
10. Roger C Schank and Robert P Abelson. 1977. Scripts, Plans, Goals and Understanding. Lawrence ErlbaumAssociates.
11. Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end Neural Coreference Resolution.In EMNLP 2017.
12. Humphreys, K., Gaizauskas, R., and Azzam, S. 1997. Event coreference for information extraction. In Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts, Madrid, Spain. Stroudsburg, PA: ACL, pp. 75–82.
13. Chen Chen and Vincent Ng. 2013. Chinese Event Coreference Resolution: Understanding the State of the Art. In Proceedings of the 6th International Joint Conference on Natural Language Processing, number October, pages 822–828.
14. Naushad UzZaman. 2012. Interpreting the Temporal Aspects of Language. Ph.D. thesis, University of Rochester. Zeno Vendler. 1957. Verbs and times. The Philosophical Review, 66(2):143–160.
15. Karl Pichotta and Raymond J. Mooney. 2016. Using Sentence-Level LSTM Language Models for Script Inference. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 279–289.
16. James Pustejovsky, Patrick Hanks, Roser Saur´ı, Andrew See, Robert Gaizauskas, Andrea Setzer, Beth Sundheim, David Day, Lisa Ferro, and Dragomir. 2002. The TIMEBANK Corpus. Natural Language Processing and Information Systems, 4592:647–656
17. Llorens, H., Saquete, E., and Navarro-Colorado, B. 2013. Applying semantic knowledge to the automatic processing of temporal expressions and events in natural language. Information Processing & Management 49(1): 179–97.
18. Vanitha Guda, Suresh Sanampudi_© Springer Nature Singapore Pte Ltd. 2017. H.S.Saini et al. (eds.),"Pattern Based rules for times" Innovation in Computer Science and Engineering, Lecture Notes in Networks and Systems 8, DOI 10.1007/978-981-10-3818-1_6
19. James Allen. 1984. Towards a general theory of action and time. Artificial Intelligence, 23:123–154.
20. Vanitha Guda, Suresh Sanampudi Springer Nature Singapore Pte Ltd. 2018. "A Hybrid Method for Extraction of Events from Natural Language Text" S.C. Satapathy et al. (eds.), Data Engineering and Intelligent Computing, Advances in Intelligent Systems and Computing 542, DOI 10.1007/978-981-10-3223-3_28.

AUTHORS PROFILE

Mrs Vanitha Guda Assistant professor at Chaithanya Bharathi Institute Of Technology(A), 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. His area of specialization includes Natural Language Processing, Text summaries, etc. He published nearly 20 publications in various journals and conferences.