A NOVEL MODEL FOR TIMED EVENT EXTRACTION AND TEMPORAL REASONING IN LEGAL TEXT DOCUMENTS

Kolikipogu Ramakrishna[1], Vanitha Guda[2], Dr.B.Padmaja Rani[3] , Vinaya Ch[4]

[1]Department of Computer Science & Engineering, CMRCET, Hyderabad, A.P, India
krkrishna.csit@gmail.com
[2,4]Department of Computer Science & Engineering, CBIT, Hyderabad, A.P, India
vanithaguda@gmail.com,vinayachakati@gmail.com
[3]Department of Computer Science & Engineering, COE, JNTU, Hyderabad, A.P, India
padmaja_jntuh@yahoo.co.in

ABSTRACT

Information Retrieval is in a nascent stage to provide any type of information queried by naïve user. Question Answering System is one such successful area of Information retrieval. Legal Documents (case law, statute or transactional document) are increasing day by day with the new applications (Mobile transactions, Medical Diagnosis reports, law cases etc.) in the world. Documentation of various Business and Human Resource (HR) applications involve Legal documents. Analysis and temporal reasoning of such documents is a demanding area of research. In this paper we build a novel model for timed event extraction and temporal reasoning in legal text documents. This paper mainly works on "how one can do further reasoning with the extracted temporal information". Exploring temporal information in legal text documents is an important task to support legal practitioner lawyer, in order to determine temporal based context decisions. Legal documents are available in different natural languages; hence it uses NLP System for pre-processing steps, Temporal constraint structure for temporal expressions, associated tagger, Post-Processor with a knowledge-based sub system helps in discovering implicit information. The resultant information resolves temporal expressions and deals with issues such as granularity, vagueness, and a reasoning mechanism which models the temporal constraint satisfaction network.

KEYWORDS

Event; Time; Temporal reasoning; legal documents; temporal tagger; temporal Constraint network;

1. INTRODUCTION

Temporal information is most ubiquitous in legal text documents to make some important conclusions in the context of decision making. Reasoning of temporal information is upcoming research area; there is lot of scope to enhance it to the down levels of Specific Events Extraction with temporal reasoning in legal text documents. The legal documents are increasing day to day including criminal laws, commercial laws, labour laws, transactional documents etc. in to the huge repositories. For instance ordering and linking of information in legal text documents with temporal relation has become essential in all the public and private sectors. It is difficult to provide computational adequacy for the temporal information in automatic extraction, representation and reasoning in the legal text. Temporal representation and reasoning theories are drawn from many domains like philosophy, computer science, linguistics and cognitive science. Temporal logics and ontologies have been implemented in many systems with different expressive power and computational complexity. Apart from the temporal reasoning plan Recognition [5] is also essential in this context. Question Answering systems, Text Summarization, Medical diagnostic reports, determining consistency satisfaction between all temporal variables, deducing new relations (computing their closure). Temporal extraction and

DOI : 10.5121/ijcses.2011.2103

reasoning are required in human life circle when he or she involved in a legal case. In general the search for previous cases relevant to the current case, that may not be superseded by decisions of a higher court. A viable alternative to the classic approach of ordering legal cases according to a time line, there are other applications where the automatic temporal ordering of documents becomes more important for a legal document search. Temporal representation and reasoning in legal text documents using natural language techniques are significant tasks because of: diversity of time expressions, complexity of determining temporal relations among events, difficult of handling temporal granularity, other major problems in computational NLP(E.g.:- ambiguity, anaphora, ellipsis and conjunction). To analyze automatic temporal information process, one has to know how temporal information is transformed into text. In this paper we examine the aspects of existing NLP systems that need to be improved to process temporal data. We investigate and evaluate suitable temporal ontologies and reasoning mechanisms for enhancements over the system. We proposed a novel model for extracting, reasoning and representation of temporal information in legal text documents. The research is done to build a framework that supports further application which helps legal practitioners and researchers to detect inconsistencies between events and witness statements. The rest of the paper is organized as follows: In section 2 abstract frameworks is provided to have a glance over the entire process. Section 3 describes individual components. In Section 4 implementation difficulties and practical approach is narrated. At Last in conclusion the advantages, potential disadvantages of the model and future work directions to enhance the proposed model are discussed.

2. FRAMEWORK

The System Architecture is shown in figure1. The individual components of a framework integrated with: 1.NLP system. 2. Annotation structure and tagger for temporal expressions and events. 3. Post processor including a knowledge-based sub system and 4. A reasoning mechanism which models temporal events in temporal constraint satisfaction networks (TCSNS).

2.1. Natural Language Processing

The NLP system functional components includes: 1. Sentence Splitter. 2. Tokenizer. 3. Shallow parser. Initially NLP System takes legal text documents as an input and then it performs processing using POS Tagger (parts-of-speech). Sentence splitting and tokenization is done by POS Tagger. Output of POS Tagger is passed to the Name identification module it updates corresponding named entity tags with semantic information and gender based on application. **S**entence Splitter first marks individual sentences in the text by using a set of heuristic rules for detecting the end of the sentences. The sentence boundaries are identified by scanning the text i.e. sequence of characters separated by white spaces (tokens) containing one of the symbols like'.',',',!'. **T**okenization module splits the individual sentences into words using white space as delimiter and removes stop word list from the sentence(stop words like ***this, is, a, for, by ,*** etc.., prepositions, conjunctions, articles etc.). The system uses Shallow Parsing method to locate interesting passages and sentences from the retrieved documents by using keyword based techniques. It also filters the passages and sentences based on the presence of the desired answer type within that candidate text. Shallow or Partial parsing is a task of recovering only a limited amount of semantic or syntactic information from the input natural language (NL).

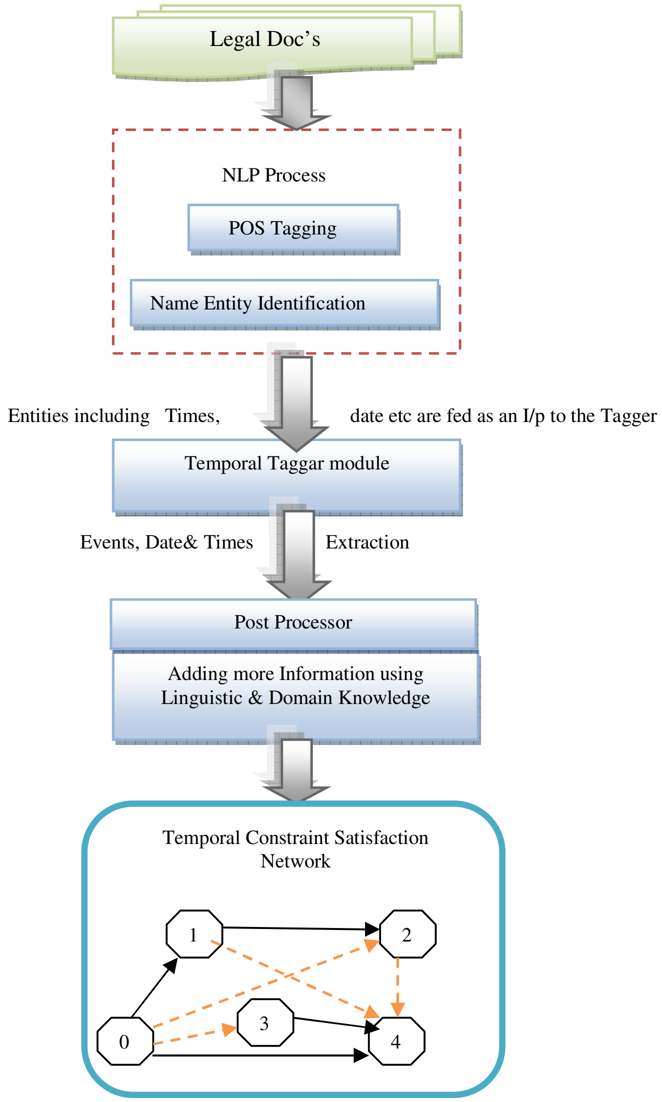
Legal Doc's
NLP Process
POS Tagging
Name Entity Identification
Entities including Times, date etc are fed as an I/p to the Tagger
Temporal Taggar module
Events, Date& Times Extraction
Post Processor
Adding more Information using Linguistic & Domain Knowledge
Temporal Constraint Satisfaction Network
1
2
3
0
4

Figure 1: System Architecture

2.2. Annotation and Tagging of Temporal information

The specification language TimeML [6][7] offers a guidelines for the annotation of temporal and event information. Similar to our approach, TimeML keeps the representation of temporal expressions and events separation: temporal expressions are tagged by TIMEX3, whereas events are annotated by EVENT. In addition two further tags called SIGNAL and LINK are used. Events are annotated by the <EVENT> tag. Events are described by the situations that are happened or occurred. It includes states which are situations in which something holds true. It annotates additional information, such as attributes, encompasses the event class (e.g. OCCURRENCE or STATE), the tense and the aspect (e.g. PROGRESSIVE or PERFECTIVE). Temporal information 12:00***, 24. December 2002* or *3 days*** is tagged by <TIMEX3>. The definition of TIMEX3> is based on the TIMEX specification given by Ferro et al.[13] and James Pustejovsky[12]. Function words that signals the relation between temporal objects that are tagged by <SIGNAL>. In particular temporal prepositions and connectives indicate a relation (e.g. before, during or when). This tag was first introduced by Setzer [12]. With TimeML, polarity indicators (e.g. ***no, not***) and temporal quantification words (e.g. ***Twice, thrice***) are also annotated as signals. The <LINK> tag specifies the exact relation that holds between the temporal elements in the text. The <LINK> tag either indicates a temporal (i.e. TLINK), a subordination (i.e. SLINK) or an aspectual (i.e. ALINK) relation. Annotation scheme identifies timers and also events in the text document. TimeML keeps the representation of temporal expressions and events separate; temporal expressions are tagged by TIMEX3, where as events are annotated by event. In addition, two further tags are used: SINGAL and LINK. For instance a sentence like "***James left two days before the attack***" is annotated with TimeML as shown below. The annotation process extracts two events i.e., left and attack), one temporal expression (i.e., 2 days) and one signal word (i.e., before).In addition, two event instances ei$_1$ and ei$_2$ are annotated. The links are established between these events is signalled by the temporal preposition before. Hence a temporal relation is stated in the TLINK tag. The attributes for the tags are not of any importance for the remainder of the paper. The usage of the attributes is explained in more detail in [2].

Example1:

James

<EVENTeid="ei1"class="OCCURRENCE" tense="PAST" aspect="PERFECTION">

Left

</EVENT>

<MAKEINSTANCE eid="ei1" event id="ei1"/>

<TIMEX3 tid="t1" type="DURATION" value="P2D" temporal function="false">

2 days

</TIMEX3>

<SIGNAL Sid="s1">

before

</SIGNAL>

the

<EVENTeid="e2"class="OCCURRENCE" tense="NONE" aspect="NONE">

attack

</EVENT>

<MAKEINSTANCE eiid="ei2" event id="e2"/>

<TLINK eventinstanceID="ei1" signaled="s1" relatedtoevent="ei2" reltype="BEFORE" magnitude="t1"/>

We model the time over which an event occurs as an interval. Each interval has start and finishes time points, each of which may be constrained by temporal expressions. For example, from the statement "***The John was a famous killer until 1984***".We can infer that John stopped killing in 1984.However we do not know when he started killing. We represent temporal expression by placing the end points of an event in a relative timeline and determining the relation or metric relationship between the endpoint and its anchor. E.g., the John stopped killing (the finish) in (equal) 1984 (the anchor).

Example2:

Master wants to teach on Monday.

Master

<EVENT eid="e1" class="I_STATE" tense="PRESENT" aspect="NONE">

wants

</EVENT>

```
<MAKEINSTANCE eiid="ei1" eventID="e1"/>
<SLINK eventInstanceID="ei1" signalID="s1" subordinatedEvent="e2" relType="MODAL"/>
<SIGNAL sid="s1">
to
</SIGNAL>
<EVENT eid="e2" class="OCCURRENCE" tense="NONE" aspect="NONE">
teach
</EVENT>
<MAKEINSTANCE eiid="ei2" eventID="e2"/>
<SIGNAL sid="s2">
on
</SIGNAL>
<TIMEX3 tid="t1" type="DATE" temporalFunction="true" value="XXXX-WXX-1">
Monday
</TIMEX3>
<TLINK eventInstanceID= “ei2" relatedToTime="t1" relType="IS_INCLUDED"/>
```

Temporal and event-based reasoning in natural language text, particularly applied to information extraction and reasoning tasks [14] [15] [16]. Several papers from the workshop [16] point to promising directions for time representation and identification. Many issues relating to temporal and event identification remain unresolved, however, and it is these issues that TimeML was designed to address. Specifically, four basic problems in event-temporal identification are addressed:

(a) Time stamping of events (identifying an event and anchoring it in time);

(b) Ordering events with respect to one another (lexical versus discourse properties of ordering);

(c) Reasoning with contextually underspecified temporal expressions (temporal functions such as last week and two weeks before);

(d) Reasoning about the persistence of events (how long does an event or the outcome of an event last).

The specification language, **TimeML**, is designed to address these issues, in addition to handle basic tense and aspect features. The temporal constraints network model contains a set of fields that are ***start*** and ***finish*** constraints related to events. The definition of each fields, possible values and examples are described in ***table 1.***

2.3. Post Processor

The role of the post processor is to implement issues including temporal granularity, implicit and explicit vagueness, and uncertainty. Apart from this the post processor is to resolve time expressions especially indexical expressions (e.g., ***“now”,*** and “***today”***), which designates times that are dependent on the temporal context. Data from other sources helps to solve this problem. The most of the temporal information extracted in the first two steps is explicit; but the implicit kind of information can be extracted by the post processor. In turn there a challenge in use of processed data in NLP System, because the NLP system has to deal with the temporal concepts may not always be linked. In order to link time to event or one event to another, we built a knowledge-based subsystem by utilizing different sources. In this structure the major modules are linguistic knowledge, Domain specific knowledge and modelling temporal constraint network.

2.3.1. Linguistic Knowledge

Temporal discourse analysis helps to order events and link them to time. One default rule that we can use is “narrative time progression”, which means “except when explicitly contradicted in the text, each successive statement either has the same "event-time" as the event in the preceding statement or is temporally after it, but is never before it". We studied that these rules can be built based on discourse structure, lexical knowledge and syntax. All these are to override the default "narrative time progression".

2.3.2. Domain Specific Knowledge

Time and Actions are both ubiquitous in legal domains. Notations related to time are found in major legal areas such as labour laws where time conditions to compute benefit periods. Commercial laws where as the time information used to establish validity of agreements, Criminal laws documents where the temporal information known about the various elements involved in the analysis of a criminal case. Our goal is to provide a representation framework well-suited to formalizing the temporal aspects of law in different areas. Most commonly documents handled by the lawyers (legal practitioners) in their daily work include transactional documents. That includes contracts, purchase, sales agreements and others which represent some kind of legal transaction. These documents almost contain time expressions important for the legal stature of the document. We are built a system to recognize these dates in transactional documents. Dates need to be fully defined in this sense and so a reader is never required to infer the year or month based on other evidence in the document. We found various types of dates appear in the transactional documents.

Example 1:

January 31, 2002
15th day of January, 2002
15/07/1985 (dd/mm/yyyy)
10/27/1994 (mm/dd/yyyy)
21-07-1989 (dd-mm-yyyy)
30.08.1998
The 24<sup> th </sup> day of January 2002
January-------, 2002
this--------day of January, 2002
first (1st) day of June, 2002
this 25th day of August, 2002
In this system, to identify all the above stated date types we use recursive descent parser to extract quantitative dates from the documents.

2.3.3. Modelling Temporal Constraint Network

After extracting, representation of the temporal information and events from the actual legal text documents is an appropriate computational mechanism needed for automatic and efficient reasoning of temporal relationships. In this System we applied constraint propagation techniques i.e. graph-based representation where vertices represent event times and edges represent the possible temporal relationships. This is shown in figure 2. This graphical notation represents vertices as intervals or points and the directed edges as labels. These labels are added with elements from the appropriate algebra representation the disjoints the possible relations between the two intervals or points. To leave in a consistent sate the graph is a labelled with basic relations. Such labels represented as singleton set (a set of single basic relations) and it is possible to map the vertices to a time line which have the single relations between vertices that it holds. The set of feasible relations between two vertices consists of only elements (basic relations) in that label, which is capable of being part of a consistent scenario. Finding the feasible relations that it contains and removing only those elements from the labels that are not part of a consistent scenario is also important task.

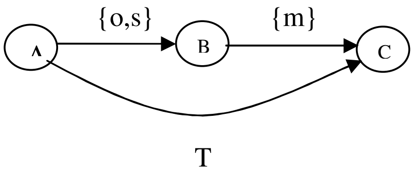

Figure 2: Temporal Constraint Network

Where the label I, the set of all basic relations, shows that there are no direct knowledge of relationship between ***A*** and ***C***. There are two possible answers to ***problem1***: find a scenario that is consistent with the information provided. Here is an example from the interval algebra for two reasoning tasks. Suppose Event ***A*** either overlaps or starts Event ***B***, but we are not sure which one that is and Event ***B*** meets Event ***C***. This scenario is represented as follows.

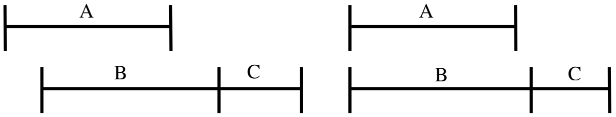

Figure 3: Events Relationship

To answer ***problem 2***: find all feasible relations between every pair of intervals. The only change is that the feasible relations between ***A*** and ***C*** is made as "***before***" relation. This truth reflects in the ***figure 3***. No other relation between ***A*** and ***C*** can be part of a consistent scenario. The following algorithm is to find the consistent scenario that takes times point network with ***n***-points. If no consistent scenario exists, the algorithm reports the inconsistency. First path consistency algorithm [3] is applied before finding a consistent scenario.

Table 1: Various fields of the Temporal Constraints

Fields	***Definition***	***Values & Examples***
Event point	Endpoint(s) of the event which is constrained by the temporal expression values	Start, finish, both or unspecified
Anchor	Constraining time point (eg. In "operation on 10/20/2003",anchor is(10/20/2003)	A calendar date, a time of day, a relative date or time an event or a time reference.
Anchor_point	If anchor is an event the end point of the event is specified	Start, finish, both or unspecified
Anchor_modifier	Indicate the stage of a period of time or the course of an event (eg. Anchor "1990's"in "late 1990's" has a modifier of "late"	Early, mid and late
Relation	A temporal relation between an end point of an event and its anchor or an interval constructed by the constraint structure with respect to anchor	Equal, before. after, equal_before, or equal_or_after
Time_unit	Unit for measuring time period	Year, month, day, hour, etc.
Quantity	Specified or indefinite number or amount for measuring the length of a time period	A number or a vague quantifier (Eg: many)
Direction	Indicate the direction of an interval relative to its anchor	Plus (future), minus (past),or both. (eg; within 3 weeks)

Interval_operator	Characterizes an end point of an event; determines whether an end point of an event occurred a specified duration away from the anchor(jump)or any time between the anchor and a specified duration away from the anchor(drag)	Jump(eg 3 weeks ago)or drag(eg: within the past 2 month)
Vagueness	Indicates if a vagueness modifier is contained within the expression (eg about in"about 2 weeks ago" approximately.	Yes

3. Consistent Scenario Algorithm for PA Network

Input: A Point Algebra [PA] network represented as a matrix ***C*** where element ***Cij*** is the set of possible relations on ***edge (i, j).***

Output: A consistent scenario (a linear ordering of the points).

Step 1: Identify all the strongly connected components (SCCs) of the graph using only the edges labeled with <, ≤, and =.Condense the graph by collapsing each strongly connected components into a single vertex. Let {S1,S2,…Sm} be the SCCs we have found (the Si partition the vertices in the graph in that each vertex is in one and only one of the Si).

We construct the condensed graph and its matrix representation, C, as follows. Each Si is a vertex in the graph. The labels on the edges between all pairs of vertices is given by

$$C_{ij} \leftarrow \bigcap_{\substack{v \,\text{€}\, S_i \\ w \,\text{€}\, S_j}} C_{vw}, \quad I, j=1,\ldots,m$$

if the empty label, ø, results on any edge, then the network is inconsistent.

Step 2: Replace any remaining ≤ labels in C with <. Perform a topological sort using only the edges in C labelled with <.

4. Implementation & challenges

In this paper we described four major components of in the framework that are 1) NLP System. 2) Temporal tagger. 3) Post Processor. And 4) Temporal Constraint Network. As completely as possible we presented temporal constraint structure that describes the original meaning of temporal information. TimeML group developed a temporal annotation guideline; it mainly focuses on news articles domain. Legal domain is more important now a days, so it is necessary to extend scope of natural language techniques to perform temporal reasoning in legal domain. In this domain legal reasoning deal normally in ad-hoc basis. Traditionally legal reasoning has been focused on AI-related research area, where content of laws and regulations may appear. Apart from this, in order to perform legal reasoning sufficient capabilities are required such as:

- Ability to reason with cases (Legal case) and examples particularly through analogy.
- Ability to handle ill defined and open textured predicates.
- The ability to handle exceptions.
- Ability to handle fundamental conflict rules.
- Ability to handle change and non-monotonicity.

Case Laws are emerging area for AI researchers for several reasons. Artificial Intelligence can act as "*A solution to the problem of legal problems of legal complexity*". Computer programs can indeed solve legal problems. The fact that computer programs can model law is not adequate simply of academic interest. Automated case research is one potential application of intelligence which determines a solution to legal problems, it can automatically fetch relevant cases from online or offline statutory and case law databases. This framework is implemented in Python Scripting language which provides object oriented features for IR applications as business logic language. XML is used as intermediate language to provide semantic web. The system implements all operations described in this paper and uses Application Programming Interface developed with HTML Technology. To measure the effectiveness of the proposed system, we experiment our model with few legal documents belonging to three categories including *Buying-selling document, Enterprise foundation document, Bank loan document*s. The results are plotted using statistical measurements called Precision and Recall defined as follows:

$$\textit{Recall} = \frac{R}{R + RNR} \; X\,100; \qquad \textit{Precision} = \frac{R}{R + NR} \; X\,100;$$

R is a set of retrieved features, ***RNR*** is the number of relevant features, but Not Retrieved and ***NR*** is the number of Non Relevant features retrieved by the system.

This paper is our first attempt to know how temporal information extraction techniques can be carried with formal temporal reasoning approaches. We would like to extract parts of legal text interested by legal practitioners (lawyers) with Information Extraction (IE) methods like finite state transducer. Event extraction is first task when a lawyer is interested in a specified event type mentioned in various legal documents as statutes. In second task it captures the temporal constraints that may be associated with the extracted events. As a resulting step model the temporal constraint network for reasoning the information. Legal Repositories are scattered in different locations. The autonomous structure of local system need to be integrated with distributed approach. Once the crawlers are allowed to crewel the legal document as and when required it is possible build such model that works well in the networked environment. Recognizing the different time formats over existing documents is also a challenging task that may comes with Natural languages. Finding timed events like “October 2nd 1982 “are also difficult. There a necessity to develop new tools that recognizes the time events in all the formats. It is also a challenge to the researchers to develop linked format network to the time formats.

5. CONCLUSION AND FUTURE WORK

The proposed framework processes the time oriented information in legal text documents. It helps to find the multiple legal attempts if any based on time event. Most of the temporal reasoning systems limit to analyze the past behaviour for decision making. Those fail in finding multiple cases on same timestamps. This attracted to propose a novel approach that addresses the timed event extraction and reasoning. This model starts with NLP prepossessing techniques followed by multiple knowledge bases, and a temporal reasoning formalism. This paper aims to support legal practitioner lawyers by providing temporal relationships for decision support. The Current model can be even implemented for Domain specific Event Extraction and Reasoning. This can be also adapted to crime investigation in various Fields including Online Fraud Detection, Cell Phoning Crime investigation etc.

ACKNOWLEDGEMENT

Authors thank Mr.M.V.Naik for his technical support to implement this model. They also thank family members, friends and college management for supporting in all the ways to continue collaborative research in the field of Information Retrieval.

REFERENCES

[1] Anne vonder Lieth Gardner: An Artificial Intelligence Approach to Legal Reasoning, 1988.

[2] A.Gerevini,L.Schubert, andS.Schaeffer."Temporal Reasoning in Time Graph", I-II.SIGART Bulletin, 4(3):pp 21-25, July 1993.

[3] Dechter.R,Neiri.I, Pearl.J," Temporal Constraint Networks. Artificial Intelligence", JAMIA,:49:91-95,1991.

[4] Frank Schilder: Event Extraction and Temporal Reasoning in Legal Documents, LNAI 4795,pp. 59 71, 2007@ Springer-Verlag Berlin Heidelberg 2007 .

[5] H.Kautz. "A Formal Theory of Plan Recognation and It's Implementation In J.F Allen, H.Kautz R.Pelavin,J.Tenenberg, Editors, Reasoning about Plans,San Mateo,CA,1991.Morgan-Kauefmann.

[6] James Pustejovsky et al, 2004',, "Specification Language TIMEML", The Language of time: A Reader, Oxford University Press.

[7] James Pustejovsky et al, 2003' "TimeML Robust Specification of Events and Temporal Expressions in text" in proceedings of AAA1 Spring Symposium on New Directions in Question answering

[8] James Pustejovsky,Sauri,R.Setzer,A.Ingria,B."Time Ml Annotation and Guidelines",TERQAS report,2002

[9] Ladkin, P. B., and R. Maddux. 1988. The Algebra of Constraint Satisfaction Problems and Temporal Reasoning. Technical Report, Kestrel Institute, Palo Alto, Calif

[10] Schillder, F., McCulloh, A.: Temporal information extraction from legal Documents/2005/313.

[11] Vila, L., Yoshino, H.: Time in automated legal reasoning. Information and Communications Technology Law 7, 173-197

[12] James Pustejovsky, The TIMEBANK Corpus

[13] Mani, James Pustejovsky, Robert Gaizauskas – 2005," The language of time: areader"

[14] Inderjeet Mani, Machine Learning of Temporal Relations,ACL 2000.

[15] Schiffman, B., Mani, I., and Concepcion, C. (2001). Producing Biographical Summaries: Combining Linguistic Knowledge with Corpus Statistics. In Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'2001)

[16] Mani Inderjeet, Annotation Standards for Temporal Information in Natural Language, LREC 2002

Authors

1. ***Kolikipogu Ramakrishna*** received B.Tech (CSIT) from VCE (JNTUH), M.Tech in Software Engineering from GIET-JNTU Kakinada. Currently he is working as Associate Professor in CSE Dept, CMRCET, Hyderabad. He is also a member of ISTE, IACSIT and IAENG; He is Pursuing Ph.D in Computer Science & Engineering in the area of Information Retrieval in JNTU Hyderabad. He has 7 years of Professional teaching experience in various Engineering colleges. For his credit he has published 2 IEEE International National, 2 national Conference Papers, 2 International Journals.

2. ***Vanitha Guda***, M.Tech (CSE) is working as Assistant Professor in CSE Department, Chaitanya Bharathi Institute of Technology. She has 4 + years of Teaching Experience in Engineering. She is distinguished Resource person for Various MNCs. Her Area of interest is Automata and Compiler Design, Information Retrieval Systems, NLP and Machine Translation.

3. ***Dr.B.Padmaja Rani*** received B.Tech(ECE) from Osmani University, M.Tech(CSE) from JNTU Hyderabad and Ph.D from Jawaharlal Nehru Technological University. She is a member of ISTE, CSI; she is currently working as Associate Professor in Dept. of CSE, College of Engineering, JNTU Hyderabad. She is having 11 + years of Professional Teaching Experience and 5+ years of research experience. To her credit she is having 25 international Journals/Conference Papers. Her area of Research interest includes Information Retrieval in Indian languages, NLP.

4. ***Vinaya Chakati***, M.Tech (CSE) is working as Assistant Professor in CSE Department, Chaitanya Bharathi Institute of Technology. She has 4+ years of Teaching Experience in Engineering .Her area of interest is Information Retrieval System, Artificial Intelligence.