Reasoning in Legal Text Documents with Extracted Event Information

Venkateswrlu Naik. M
JNT University
Hyderabad
CMRCET, Hyderabad

Vanitha Guda
Osmaniya University
Hyderabad
CBIT, Gandipet, Hyderabad

Inturi Srujana
Osmania University
Hyderabad
CBIT, Gandipet, Hyderabad

ABSTRACT
Extracting Events, Time Expressions and Named Entities from Legal text is fundamental aspect for deep language understanding and key to various applications such as Temporal Reasoning in Criminal Documents, Case decisions(Intellectual property and crime) for details, Case Based Reasoning, Ordering of Cases according to their Time lines, Determining Relevancy between Precedent cases and Current cases, Temporal Question Answering System, Text Summarization and Documents Retrieval according to Events and Times. Our long term intension is to build a system which automatically extracts Events and Time expressions and ordering them in a particular order. Ordering of events become significant task and it is assists to finding all feasible times a given event can occur, all relationships between two given events, finding one or more consistent scenarios and finally representing data in a minimal network form.

In this paper, we are focusing about automatic extraction of Quantitative, Qualitative time's information and from Legal Text Documents, along with this Legal text expressed in natural language can be automatically annotated with semantic mark ups using natural language processing Techniques. Finally applied reasoning among temporal information with the help of extracted information. Reasoning can be done using constraint satisfaction networks by applying Allen's Algebra relations. Apart from this result analysis obtained using **Precision** and **Recall** statistical measurements over standard dataset DUC 2005.

General Terms
Tokenization, Parts of speech tagging, Named Entity Recognition, Relation Recognition between Events, Time Extraction, Event Extraction. Quantitative times.

Keywords
Qualitative time's, Time Extraction, Time Markup Language (TIMEML), Event Extraction, Legal text documents, Temporal Reasoning, Semantic Representation.

1. INTRODUCTION
The amount of natural language text that is available in electronic form is truly staggering, and is increasing every day. However, the complexity of natural language can make it very difficult to access the information in that text. The state of the art in NLP is still a long way from being able to build general-purpose representations of meaning from unrestricted text. If we instead focus our efforts on a limited set of questions or "entity relations," such as "where are different facilities located" or "who is employed by what company," we can make significant progress. The goal of this paper is to answer the following questions:

1. How can we build a system that extracts structured data from unstructured text?

2. What are some robust methods for identifying the entities and relationships described in a text?

Temporal information is most ubiquitous in legal text documents to make some of the important conclusions in the context of decision making. Reasoning with temporal information has attracted great attention due to its availability in various legal documents such as criminal laws, commercial laws, labor laws, transactional documents etc.., For instance ordering and linking of information in legal text documents with temporal relation has become essential in the dynamic world. The crucial step towards computational adequacy of the temporal information in these areas is lies in automatic extraction, representation and reasoning with temporal information described in the legal text.

Temporal representation and reasoning theories draw from many fields including philosophy, computer science, linguistics and cognitive science. Temporal logics and ontologies have been widely discussed and many systems have been proposed with different expressive power and computational complexity. Apart from the temporal reasoning essential in several areas such as Plan Recognition [1], Question Answering Systems [2][3], Text Summarization [4], Medical diagnostic reports, determining consistency satisfaction between all temporal variables involved [5], deducing new relation from those that are known(computing their closure).Temporal extraction and reasoning can be formed in our normal life circle of a case. Traditionally the search for precedent cases relevant to the current cases that are also not superseded by decisions of a higher court made at a later date [6]. Apart from the classic case of ordering legal cases according to a time line, these are other applications where the automatic temporal ordering of documents can become crucial for a legal researcher. Temporal representation [7] and reasoning in legal text documents with natural language technique is a significant task 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 understand how to automatically handle temporal information, it is first necessary to analyze how temporal information is conveyed in text, to examine which aspects of existing NLP systems need to be improved to process

temporal data and to investigate and evaluate suitable temporal ontologies and reasoning mechanisms.

Our objective is to build a model for temporal information in legal text documents includes extraction, representation and reasoning. The goal is to initiate and build a foundation that supports further application which assists to legal practitioner and research such as detection of inconsistencies between events, witness statements, and Question answering on Temporal information. The rest of the paper is organized as follows section 2 possess Information Extraction Pipe Line Model. Section 3 we describe in detail about Architectural Proposal of the individual components, section 4 discusses Temporal parsing and lastly, we discuss the application areas of our model and the future work directions to enhance the proposed model.

2. INFORMATION EXTRACTION PIPE LINE MODEL

The architecture describes regarding simple information extraction system. It begins by processing a document using several of the procedures discussed in architecture. And first, the raw text of the document is split into sentences using a sentence segmenter, and each sentence is further subdivided into words using a tokenizer. Next, each sentence is tagged with part-of-speech tags, which will prove very helpful in the next step, **named entity recognition [8]**. In this step, we search for mentions of potentially interesting entities in each sentence. Finally, we use **relation recognition** to search for likely relations between different entities in the text.

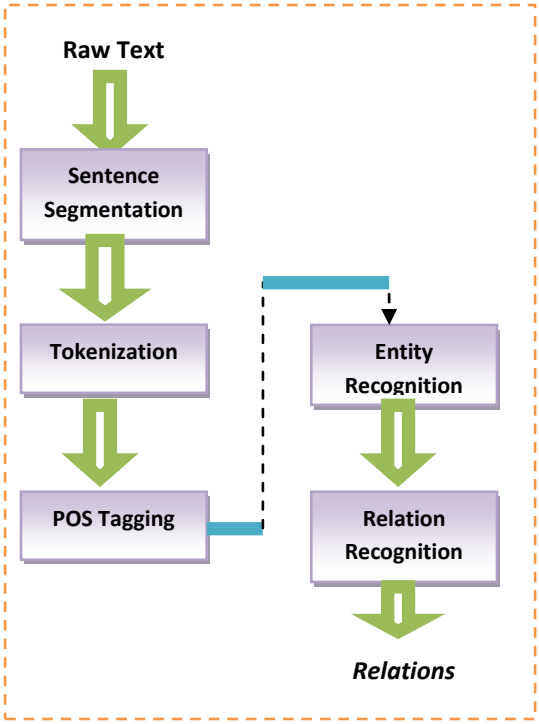

Fig 1: IE Pipe Model

3. ARCHITECTURAL PROPOSAL

In Figure 2, we propose the architecture of Automatic Extraction of Times and Events from Legal Text Documents. The model integrates key components, which are: 1) Natural Language Processing 2) Times and Events extraction 3) Temporal Parsing 4) Temporal Reasoning.

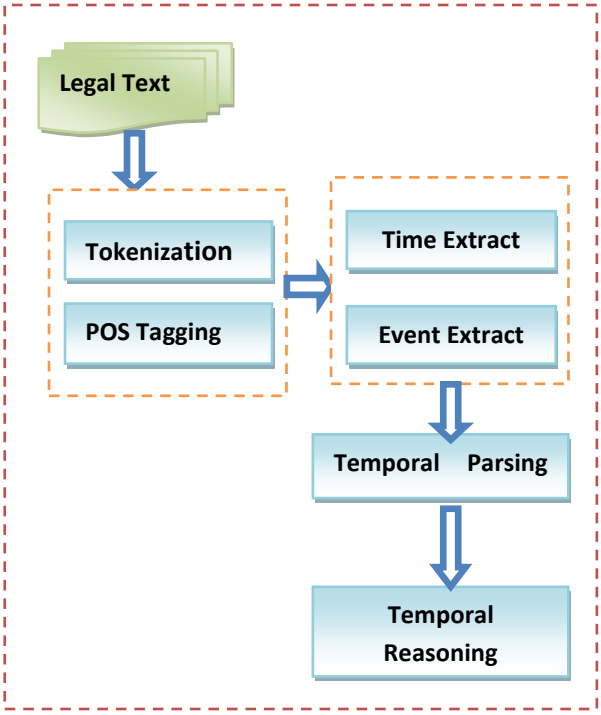

Fig 2: Architectural Proposal

3.1 NLP Processing

Natural Language Processing Module is Primary module which takes Legal Text Documents as inputs and process them by the ***Tokenization [9]*** and ***Parts-of-Speech*** [POS] Modules. **Tokenization** module determine sentence boundaries, and to separate the text into a stream of individual tokens (words) by removing extraneous punctuations. It separates the text into words by using spaces, line breaks, and other word terminators in the English language. **Parts-of-Speech Tagger [10]** assigns a part of speech label to each word in a text depending on the labels assigned to preceding words. Often, more than one part of speech tag is assigned to a single word, in turn reflecting some kind of ambiguity in the input. Its task is to assigns a syntactic category to each word in a text. The POS tags are returned in an array of the same length as the tokens array, where the tag at each index of the array matches the token found at the same index in the tokens array.

3.2 Times and Events

3.2.1 Times Extraction

Basically there are two types of **Times** appearing in the Legal text documents, namely they are i) Quantitative Times [11] (exact time formats expressions) ii) Qualitative Times [11] (reference times like today, tomorrow).Time and Actions are both ubiquitous in legal domains. Notations related to time are

found in major legal areas such as labor 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 its different areas. Most commonly documents handled by ht e lawyers in their daily work include transactional documents. These include contracts, purchase, sales agreements and other which represent some kind of legal transaction. These documents almost contain time expressions important for the legal stature of the document. We are building 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. Some of the examples can be seen here.

The J**anuary 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 24th 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 used generic pseudo code using Recursive Descent parser using Context Free Grammar to extract quantitative dates from the documents and conversion into standard format as below procedure.

1) Scan the first line of the document and checking for any valid date expressions.

2) If date is found extract the date, month, year and store them in separate variables.

3) For each date found print the date in the standard format i.e <DATE> <DELIMITER> <MONTH>

<DELIMITER> <YEAR>

4) Scan again from the end point of the previous date to extract any remaining date's recursively.

5) Scan the next line and continue from step1.be presented with each item marked by bullets and numbers.

3.2.2 EVENT Extraction

Event Extraction module performs two tasks majorly 1) Event Recognition with distinguish classes 2) Analysis of grammatical features such as tense and aspect. Event identification is performed based on the notations of event as defined in TimeML [10]. Various strategies have been used for recognizing events within categories of verb (active verbs, passive verbs associated with their aspects). Event identification is based on a lexical lookup, accompanied by minimal contextual parsing in order to exclude weak predicates like be or have or should or could. Identifying events expressed by nouns, on the other hand, involves a disambiguation phase in addition to lexical lookup. Time ML considers events as situations that ***happen*** or **occur**. Events can be ***punctual*** or last period of time. They consider predicates describing ***states*** or ***circumstances*** in which something obtains or holds true. Events are usually expressed as tensed or untensed verbs, nominalizations, adjectives, predicative clauses or prepositional phrases.

In addition, subordinate verbs that express events which are clearly temporally located, but whose complements are generics, are not tagged. He **said** participants are prohibited from mocking one another. Even though the verb **said** is temporally located, but is not tagged due to its complement participants are prohibited from mocking one another, is generic. As for event attributes, TIMEML use seven abstract event classes. order of subdivisions of items in bullet and numbered lists may be presented as follows:

Occurrence: die, crash, build;2) State: on board, kidnapped; 3) Reporting: say, report;4) I-Action: attempt, try, promise; 5) I-State: believe, intend, want;6) Aspectual: begin, stop, and continue; 7) Perception: see, hear, watch, feel; Apart from this, The Backus Naur Forms (BNF) rules required to tag the Event. Following lines represents BNF rules.

attributes::= eid class

eid::= e<integer>

class:: = 'REPORTING' | 'PERCEPTION' | 'ASPECTUAL' | 'I_ACTION' | 'I_STATE' |

'STATE' | 'OCCURRENCE'

Stem::= CDATA.

Fig 3 shows Legal Document Text along with ubiquitous events and all bold blue color words are extracts as events from the document.

AP-NR-08-15-90 1337EDT

Iraq's Saddam Hussein, **facing** U.S. and Arab troops at the Saudi border, today **sought peace** on another front by **promising** to **withdraw** from Iranian territory and **release** soldiers **captured** during the Iran-Iraq **war**.

Fig 3: Events in Legal Text

3.3 Temporal Parsing

Temporal Parsing overall view is illustrated in figure 4 below. Input text is first processed by the preprocessing steps, which takes care of document level properties like encoding and meta tags. Temporal parsing process posses mainly these components such as Time Tagger, Event Tagger, TLINK, SLINK, SputLINK.

3.3.1 Temporal Tagger

Time Tagger [12] is for recognizing the extents and normalized values of time expressions. This Temporal tagger can handle both absolute times (e.g., July 15, 1984) and relative times (e.g., Thursday, today) by means of a number of tests on local context. Qualitative times such as today, yesterday and tomorrow or next month, last year, when used in a specific context, these are

resolved based on local computing with respect to a reference or document publication time.

Temporal tagger tags time expressions based on the TimeML tag called <TIMEX3> which allows a functional style of encoding offsets in time in time expressions. For example, ***last month*** could be represented not only by the time value but also by an expression that could be evaluated to compute the value, namely that is the month preceding the month of the document date. Fig 5 describes how time tagger could recognize time expressions in the given example legal article.

Example Semantic Time Expression:

"three days every month" <TIMEX3 tid="t1" type="SET" value="P1M" quant="EVERY" freq="P3D"> three days every month </TIMEX3>.

3.3.2 *Event Tagger*

Event tagger is recognizing events that intended to various classes of events. Event tagger is tags events based on TimeML Tag called <EVENT> which allows a functional style of encoding of events into semantic meaning of a event along with set of related attributes. For example, "**Israel *will ask the United States to delay".***

Semantic Representation of above given example

<EVENT eid="e1" class="I_ACTION">

ask

</EVENT>

<EVENT eid="e2" class="I_ACTION">

delay

</EVENT>

In this semantic representations each event can be tagged between one pair of begin and end tags along with their corresponding attributes. These attributes can be defined in the BNF rules and TimeML tags can be validated by the DTD (Data Type Definition) or Schema's concepts.

3.3.3 *TLink*

TLINK tagger is a temporal link. It represents the relation between two temporal elements. The main purpose of the link tags is to encode the various relations that exist between the temporal elements of a document. The motivations for having multiple types of links are the following: (i) To distinguish between event types and event instances, such as those introduced by conjunction, quantification, or negation. (2) To adequately handle subordinating contexts involving modality and reported speech.

Following example represents semantic tagging of a given statements. "John taught 20 minutes every day".

John

<EVENT eid="e1" class="OCCURRENCE">taught

</EVENT>

<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PAST" aspect="NONE" negation="false"/>

<TIMEX3 tid="t1" type="DURATION" value="P20TM">

20 minutes

</TIMEX3>

<TIMEX3 tid="t2" type="SET" value="xxxx-wxx-1" quant="EVERY">everyMonday</TIMEX3><TLINK timeID="t1"relatedToTime="t2" relType="IS_INCLUDED"/><TLINK eventInstanceID="ei1" relatedToTime="t1" relType="DURING"/>

3.3.4 *SLINK*

SLINK is a subordination link that is used for contexts involving modality, evidential and fictive. An SLINK is used in cases where an event instance subordinates another event instance type. These are cases where verb takes a complement and subordinates the event instance referred to in this complement. Slink is introduced by subgroup of verbal and nominal predicates such as regret, say, promise, and attempt and most cases clearly signaled by the context of subordination.

Following example explains about semantics representation of a statement, "***John said he taught***"

<SLINK eventInstanceID ="3" subordinatedEvent="4" relType="EVIDENTIAL"/>

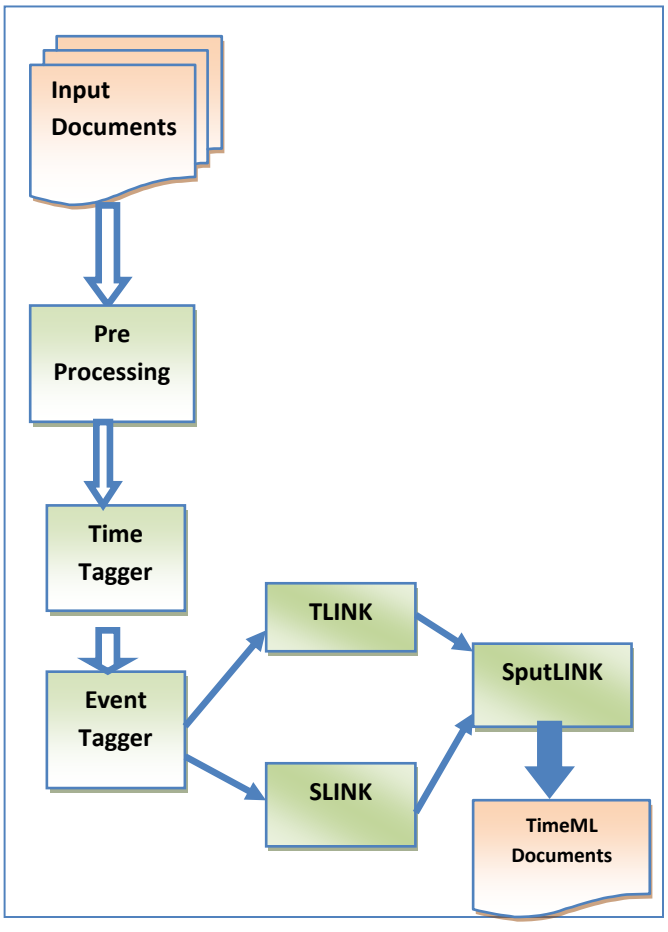

Fig 4: Temporal Parsing Model

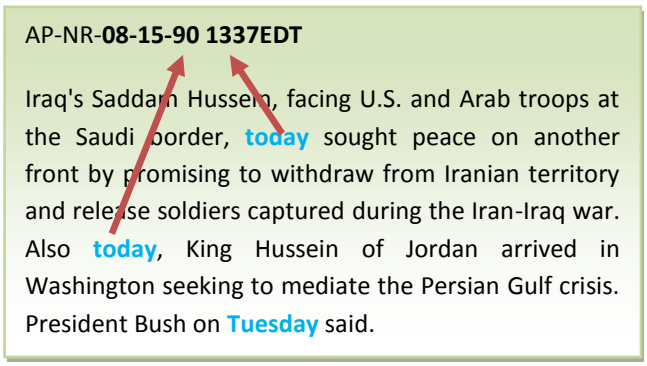

Fig 5: Time Expressions in Legal Text

4. RESULT ANALYSIS

Tables 1 and 2 show the performance results with the help of statistical measurements to evaluate the quality of developed system such as **Precision** is defined as the percentage of correct relation expressions out of recognized ones. **Recall** is the percentage of correct relation expressions from among the manually annotated ones.

In this system extraction of Time expressions qualitatively and quantitatively and Events from a particular legal text document is a statistical measures to evaluate performance of developing system by comparing other earlier system. These measures tested on standard DUC-2005 data set.

Common evaluation measures are

$$\text{Precision}=\frac{\text{No. of relevant items retrieved}}{\text{Total no. of items retrieved}}$$

$$\text{Recall}=\frac{\text{No. of relevant items retrieved}}{\text{No. of relevant items in the document}}$$

Table 1 shows observations regarding extraction of Time expressions from legal text documents by comparing developed system with earlier existing system.

Table 2 shows observations regarding extraction of Events from legal text documents from standard DUC-2005. Here Precision has been increased and Recall has been decreased when compare with earlier system.

Table 1. Evaluation of Time Expressions

System		Correct Extracted	Available Times	Percentage
Developed Extraction Model	Precision	40	44	90
	Recall	40	50	80
TempEx Model	Precision	20	24	83
	Recall	20	64	31

Table 2. Evaluation of Events

System		Correct Extracted	Available Events	Percentage
Developed Extraction Model	Precision	20	25	80
	Recall	20	32	62

5. CONCLUSIONS

We have described a model, which integrates various components that automatically extracts events, times from legal text documents and subsequently generates tags to markup events and time expressions, as well as non consuming tags that encode relations between events and times. Our model includes a module that integrates available temporal relations into a temporal constraint graph by following Allen's interval algebra which can be shown as a standard tabular form.

In this current work, we have proposed a modular model for comprehensively processing the time oriented information in terms of qualitatively and quantitatively in legal text documents. We have integrated various modules and have linked them to form a prototype model. This model integrates NLP techniques, multiple knowledge bases, and a temporal reasoning formalism. By providing a way to determine and discover temporal relationships among available events in the legal text document, our prototype assists to legal practitioner lawyers for decision support. The Current model can be even implemented for Domain specific Event Extraction and Reasoning.

This model can also adapt for future scope areas where time playing a major role and decisions must be strictly based on time. Following are the applicable areas to apply our model such as crime investigation and in various Fields including Hybrid Temporal reasoning [13], Medical Diagnosis reasoning, Online Fraud Detection, Text Summarization, Consistency Determination, and Temporal Question Answering System [14].

6. REFERENCES

[1] Naushad, UzZaman and James Allen, (2010), TRIOS-TimeBank Corpus: Extended TimeBank corpus with help of Deep Understanding of Text, to appear in the Proceedings of The Seventh International Conference on Language Resources and Evaluation (LREC), Malta.

[2] Magnini, Bernardo. Borovetz, Bulgaria: s.n, 2005 Open Domain Question Answering: Techniques, Systems and Evaluation, Conference on Recent Advances in Natural Language Processing (RANLP).

[3] Vanitha, Suresh Kumar Sanampudi, I. Lakshmi Manikyamba, 2010, Approaches for Question Answering System, IJEST, Volume no.3, Pageno:992-995, ISSN: 09755462

[4] Oi Mean Foong, Alan Oxley, Suziah Sulaiman, 2010 Challenges and Trends of Automatic Text Summarization, IJITT, Vol. 1, Issue 1, ISSN: 0976–5972.

[5] Dutcher, R, Meiri, I, Pearl, J, (1991),"Temporal Constraint Networks", Artificial Intelligence, 49:61- 95.

[6] Frank Schilder, (2007), Event Extraction and Temporal Reasoning in Legal Documents, LNAI 4795, pp. 59 7, @ Springer-Verlag Berlin Heidelberg.

[7] Inderjeet Mani, (2007), Chronoscopes: "A Theory of Underspecified Temporal Representations. Reasoning about Time and Events". Springer LNAI 4795-0127. pp: 127-139.

[8] Michael Tanenblatt, Anni Coden, Igor Sominsky, 2010 the ConceptMapper Approach to Named Entity Recognition, LREC Conference, Malta.

[9] Chu-Ren Haung, PetrSimon, Shu-Kai Hsieh, Laurent Prevot, (2007): Rethinking Chainese Word Segmentation: Tokenization, Character Classification, or Wordbreak Identification, Proceedings of the, Association for Computational Linguistics Demo and Poster Sessions, pages 69–72.

[10] Chris Biemann, 2009 Unsupervised Parts of speech Tagging in Large Text, Research on Language and Computation, Volume 7,Issue 2-4,USA

[11] Pustejovky et al, (2004), "The Specification Language TIMEML", The Language of Time: A Reader, Oxford University press.

[12] Oleksandr Kolomiyets, Marie-Francine Moens, 2009 Meeting TempEval-2: Shallow Approach for Temporal Tagger, Proceedings of the NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions, pages 52–57, Boulder, Colorado.

[13] Suresh Kumar Sanampudi, G. Vijaya Kumari, 2010,"Temporal Reasoning in Natural Language Processing: A Survey": International Journal of Computer Applications (0975-8887) Volume1-No.4.

[14] E. Saquete, P. Mart´ınez-Barco, R. Mu˜ noz, J.L. Vicedo, 2002 Splitting Complex Temporal Questions for Question Answering systems, FIT-150500-2002-244.