Temporal Reasoning in Natural Language Processing: A Survey

Suresh Kumar Sanampudi
Research Scholar
Department of CSE
J.N.T.U Hyderabad

G.Vijaya Kumari
Professor
Department of CSE
J.N.T.U Hyderabad

ABSTRACT

Reasoning with temporal information in natural language text has attracted great attention due to its potential applications in summarization, question answering and other tasks. For example, the chronological ordering of events described in a text is important for presenting the information in the summary. Linking information in a natural language text with temporal relations is essential in question answering system to address time sensitive and dynamic world. A crucial first step towards the computational treatment of the temporal information in these applications is the automatic extraction of events described in the text and identification of temporal relations to link these events. Much of the work done in this direction can be classified as -Annotation schemes for identification of events and time implicit in the text, linking the events using temporal relations and Temporal reasoning for solving practical applications.

The present paper is a survey of various proposals to address these issues. Various annotation schemas developed to represent temporal information in natural language text. A discussion of the frameworks for temporal reasoning and tractable classes is described. The usefulness of these models to applications such as summarization and question answering are also presented.

Categories and Subject Descriptors

F.1.1 [**Models for Computation**]: Computability theory, temporal logic, Logic and Constrained programming.

General Terms

Theory,Algorithms

Keywords

Temporal reasoning, temporal information extraction, time in natural language processing.

1. INTRODUCTION

Time, which is used to elucidate the changes of the world and order the events in a description is crucial in many of the natural language processing (NLP) applications such as question answering system, text summarization etc. For Instance, answering questions like ”when was John born?” requires accurate identification of the date of birth of the person under consideration (recognition) and rendering of the answer in some standard format (normalization). Text summarization techniques rely heavily on chronological coherent accounts of events.

Temporal representation and reasoning in Natural Language (NL) is a nontrivial task due to: (1) the diversity of time expressions; (2) the complexity of determining temporal relations among events; (3) the difficulty of handling temporal granularity; and (4) other major problems in computational NLP (e.g., ambiguity, anaphora, ellipsis, and conjunction). The processing of temporal information can be broadly classified into two tasks: (i) recognizing temporal expressions (timexes) that refer to point or periods of time, durations and metrics (ii) re-interpreting these temporal references in a standard format which precisely describes semantics of expressions, disambiguates dates and times from different time zones and makes it easier to determine the sequence of events described in the texts. Several works address these issues using models ranging from heavily inference oriented to mostly NL motivated ones. These models can be classified as constraint based semantic based and domain-specific models. The present work is a survey of these models with an analysis of their limitations and strengths with respect to NLP tasks.

The remaining part of the paper is organized as follows. Section 2 discusses representation and reasoning of time in natural language text. Section 3 gives a survey of constraint based models. Semantic based models are explained in section 4. Section 5 is about domain-specific model. Section 6 provides the comparison among these three models. Section 7 gives conclusion.

2. REPRESENTATION AND REASONING OF TIME IN NLP

Temporal reasoning in NLP involves extraction, representation and reasoning with time and events in the natural language text. It supports inferencing with time in solving many NLP applications. For example, story understanding, text summarization and question answering tasks in NLP are benefited greatly by temporal reasoning. Extraction and representation of time can be formalized using basic temporal primitives namely points, intervals, events (discrete or continuous), meaning of past, present, future and the relations that hold between them. These models should be suitable for representing underspecified, explicit temporal information and enable inference task to reason about time, thereby providing solution to various applications using artificial intelligence methods. A robust ability is required to denote crucial information about the facts and properties of a dynamic world. With this need there is recently renewed interest in temporal reasoning for NLP. We present a survey of various temporal models with suitable mechanisms popularly used in NLP applications.

3. CONSTRAINED BASED MODELS

Constraint based models [1][2][9] could be used in NLP. The events described in the natural language text posse's temporal aspect which can be extracted to relate the events with other events in the text. This aspect of events imposes a constraint on possible ordering of events in the NL text. The extracted temporal information can thus be represented as a temporal constraint network (TCN) with events denoted by nodes and edges by the ordering constraints of the events. Reasoning in this framework becomes a temporal constraint satisfaction problem (TCSP). TCSP is a particular class of constraint satisfaction problem (CSP) where variables represent times and constraints represent set of allowed temporal relations between them. Different TCNs are defined depending on the time entity that variable can represent namely points, time intervals, durations and the class of constraints namely qualitative, quantitative , metrics or its combination. The classes of TCSP's surveyed are Allen's interval algebra (qualitative) [1], INDU (qualitative with durations) [2], and metrics (time calculus) [8].

3.1 Qualitative Interval Algebra

Allen [1] proposed an algebraic framework called Interval Algebra (IA) for qualitative reasoning with time intervals. Using interval algebra temporal aspects of events in natural language text can be represented [10]. The pairs of intervals thereby the events in the text are related using a subset of 13 basic relations which impose ordering constraints on the intervals. The 13 atomic relations are: { before(p), meet(m), overlap (o), starts(s), finishes(f), during(d), equal(e), during-by(D), Overlaped-by(O), Started-by(S), Finished-by(F), Met-by(M), after(P)).The operators converse, composition and union are defined on the set.

The resultant temporal constraint network can be subjected to the standard reasoning algorithms for reasoning with time using path consistency [15], for propagation of constraints and generation of ordering among events in the text. For all the tractable classes in this algebra [4] consistency can be decided in polynomial time. The instantiation of time intervals to the events can also be done in polynomial time. For the remaining class of problems backtrack search could be used incorporating path-consistency as a forward checking procedure within backtrack search have been shown to be very effective in pruning dead-ends [15]. Allen's interval algebra is limited in the sense that it can only represent relative ordering of the intervals but not its duration and negation of temporal information such as "I am not reading during the next 2 hours" .

3.2 Qualitative with Durations

In [2], an elegant framework INDU, to represent temporal relations between events with qualitative durations is proposed. INDU has 25 basic relations to relate any pair of events. These relations are exhaustive and mutually exclusive. The three basic binary operations converse, composition, union are defined on it to relate qualitatively any pair of intervals. The 25 basic relations in INDU are before ($p^{<}$,$p^{=}$,$p^{>}$), after ($P^{<}$,$P^{=}$,$P^{>}$), meet ($m^{<}$,$m^{=}$,$m^{>}$), met-by ($M^{<}$,$M^{=}$,$M^{>}$), overlap ($o^{<}$,$o^{=}$,$o^{>}$), overlap-by ($O^{<}$,$O^{=}$,$O^{>}$), finishes ($f^{<}$), finished-by ($F^{>}$), starts($s^{<}$), started-by($S^{>}$), during ($d^{<}$), during-by($D^{>}$), equals($e^{=}$), These relations are capable of expressing interval duration along with endpoint relations qualitatively.

Given a natural language text using INDU framework a temporal constraint graph can be formulated for it. The nodes of these graphs represent events while the edges labeled with a subset of 25 basic relations which constrain the instantiation of the intervals with durations and end-point orderwise. Reasoning in this framework turns out to be a TCSP problem where standard algorithm such as path consistency [15] can be applied to generate a solution. Though reasoning in this framework is NP-Complete, tractable subclasses have been identified [8], which admit major classes of relations encountered in natural language text. Though INDU has a good mix of expressiveness and tractability, certain aspect of events in natural language text such as quantitative duration and semantic characteristics of temporal expressions are not feasible.

3.3 Metrics

Constraint-based frameworks for metrics adopt calendar constraint system along with set of operators and relations for processing and reasoning over temporal information in natural language. To support this framework, Time calculus [3] is used as a typed formal language for encoding natural language expressions. Time calculus includes constraint solving which derive underspecified information in the expression, operators achieving granularity change, and modeling of deitic and relative expressions.

This model identifies timex such as Coordinates (represent points in time), Quantities (represent duration of time), and Enumerations (represent a set of points). The quantitative duration can be easily represented. The basic vocabularies for composing a timex consist of temporal units and values. A temporal unit is a time metric unit, which can take on a set of possible values. For examples, "month" is a unit, and "March" is one of the possible values of the unit. Some units will have infinitely many values, such as "year". The most widely used units and values are from Gregorian calendar.

Operators and relations in Time Calculus, transparently bring all of the involved timex into the required types beforehand. A list of coordinate-producing and enumeration producing operators [3] includes +/- indicating Forward/backward fuzzy shifting (e.g., { +|1_{month}|} ("next month")), ++/-- indicating forward/backward exact shifting (e.g., { ++|1_{month}|} ("exactly one month after")), @ indicating Ordinal (e.g., {|3_{sun}|@{may}} ("the third Sunday in May")), : indicating Interval (e.g., [{wed, 15_{hour}}:{17_{hour}}] ("Wednesday from 3pm to 5pm")) , / indicating arithmetic reference (e.g., [[{16_{hour}}:{18_{hour}}]/|2_{min}|] ("every 2 minutes from 4pm to 6pm")), ^ indicating enumeration intersection and \ Indicate Enumeration difference (e.g., [{ +|0_{year}|}\[{|1_{month}|@{ +|0_{year}|}}:]], ("the rest of the year")).

In addition to the quantitative constraints imposed by the use of various operators, qualitative constraints can be expressed using various relations. Relations in time calculus are categorized as two types, value relations and object relations. The former appear inside a value term and are used to constrain temporal values e.g. "sometime between 4pm and 6pm": {(>= 16, <= 18) hour}, A complete list of value relations include (<) Less than, (>) Greater than and (=) equal to. The object relations appear at the term level inside objects, and are used to constrain the host objects e.g "sometime before Sept. 9, 1987": {b {1987_{year}, sep, 9_{day}}}. The list of object relations include before/after (b/bi), meets/met_by (m/mi), overlap/overlaped_by (o/oi), starts/started_by (s/si), finishes/finished_by (f/fi), during/during_by (d/di), equal (=), in/inverse (i/ii).

Reasoning with time in this framework results a constraint network, in which temporal assertions are represented as time difference constraints. The consistency of this constraint network is solved as a temporal constraint satisfaction problem (TCSP) [9]. The TCSP converts constraint network into a simple flow network and its consistency is derived by obtaining the minimal network using a modified all-pair-shortest-path problem. Sets of feasible assignments to the variables can then be obtained using a simple search on the edge labels of a minimal network. This can be solved by a polynomial time algorithm and are sufficient to represent primitive Allen relations, simple metric constraints and points anchored in absolute time. The quantitative and qualitative constraints are easily converted to "metric" network and solved uniformly. Tough solutions for problems in general is NP-complete, the problem generated by natural language text admit polynomial algorithms. However exact identification of tractable classes is not yet made. The "negation" of time period is still left inexpressible in this model

4. SEMANTIC BASED MODELS

Semantic based models represent time expressions in text using annotation schemes. The time expressions are words or phrases in text that refer to time. Time denoted by these expressions may be points, intervals, durations etc. common temporal expressions in natural language include phrases such as week, Sunday, in a month, recently etc. The collection of such expressions in a text describes a temporal scenario which is modeled as a semantic network. The models differ basically in the way these expressions are extracted, translated, and encoded. The various annotation schemes popularly used are TIMEX2 tagger[6], Chronos [11][14] and TIMEML [12][13]

The annotation process is decomposed into two steps: (i) marking a temporal expression in a document, and (ii) identifying the time value that the expression designates, or that the speaker intends for it to designate. The markup of temporal expressions is restricted to those timex which contain a reserved time word used in a temporal sense, called a 'lexical trigger' (day, week, weekend, now, Monday, current, future etc).

4.1 TIMEX2

TIMEX2 [6] is an annotation scheme which identifies temporal information that represents points in time (calendar dates and times-of-day), durations (period in time) and frequencies (set of time points).

TIMEX2 tag includes various attributes to identify temporal information in text. Each attribute has its own rules about what kind of values it can have. The attributes that identifies temporal information in TIMEX2 tag are: VAL (date or time of the annotated expression), MOD (Captures temporal modifiers, using values such as BEFORE, AFTER, LESS THAN, MORE THAN, EQUAL OR LESS, START, MID, END or APPROX), ANCHOR_VAL (date or time), ANCHOR_DIR (Captures the relative direction or orientation between VAL and ANCHOR VAL attributes), SET (identifies expressions denoting sets of times), NON_SPECIFIC (identifies non-specific expressions).

The annotation scheme also addresses several semantic characteristic of temporal expressions. Fuzzy expressions give a general sense of meaning but cannot confidently posses a precise value. VAL and MOD attributes are used to represent fuzziness in time, Non specific temporal expressions do not indicate a specific time. NON_SPECIFIC attribute is used to indicate this category of temporal information. Non specific temporal expressions fall into two categories namely Generic (specifying a whole class of temporal entities rather than referring to a specific time), and indefinites.

The temporal information identified in the recognition stage is interpreted to transform local semantic representation into a document-internal semantic representation [5]. The representation of these temporal expressions is unification based with respect to a reference date and arithmetic involving specified tme quantity with respect to this date. The interpretation is also influenced by the nature of the temporal information namely fully specified point in time, underspecified point in time, relative expression, duration, frequency and so on.

The interpretation task is composed of five major stages. Stage 1 is a lexical lookup to recognize names and temporal units in the input to map them to ISO values. Stage 2 combines the lexical tokens from stage 1 to form meaningful context independent time expressions (timexes). Stage 3 disambiguates the meaning of temporal expressions based on the context information consequently problem such as Point-duration problem (distinguishing whether an ambiguous unit phrase refers to a point or a duration), Direction problem (determining whether an ambiguous anaphoric point-referring phrase refers to a point before, after, or the same as the reference time), Today problem (determining whether an occurrence of the word today refers specifically to the day of the article or broadcast or generically to the present) are addressed. Stage 4 is focused on tracking and identifying reference time s and temporal focus. It instantiates anaphoric timexes and propagates reference time among the expressions. Finally stage 5 computes a normalized value by combining the results of all these steps. The annotation scheme however has no mechanism for representing non quantifiable durations and localizers such as as-of, during, starting, ending etc. Human interpretation is inevitable in such situations.

4.2 Chronos

Chronos [9][12] is a more complex system designed to perform both recognition and normalization of temporal expressions. Chronos system is designed to provide the automatic annotation of textual data with the TIMEX2 tag, which includes attributes for expressing the normalized, intended meaning or value of a broad range of temporal expressions.

Chronos system relies on two main components: the detection and bracketing component, and the normalization component. Detection refers to systems capability to recognize time expressions within an input text. Bracketing concerns systems capability to correctly determine the extension of a detected time expression. Normalization refers to the ability of the system to correctly assign the normalization attribute values for all the correctly detected time expressions.

The detection and bracketing component is in charge of the linguistic analysis of the input text, and the production of an intermediate annotation of such text. Text processing in Chronos involves tokenization, statistical part-of-speech tagging and multiwords recognition based on a list of 5000 entries retrieved from WordNet. Then, the text is processed by a set of approximately 1000 basic rules that recognize temporal constructions and gather information about them that is expected to be useful in the process of normalization. This is followed by the application of composition rules, which resolve ambiguities when multiple tag placements are possible.

The intermediate annotation contains all the relevant information required in the normalization phases. The normalization component exploits the intermediate annotation to assign correct values to the TIMEX2 attributes of each detected time expression. This process is carried out in three steps: anchors selection which connects detected relative expressions to absolute time expressions, dates normalization which fills VAL attribute of each detected time expression, and attributes normalization that incharge of producing the final tagged text.

At the level of rules, extensions and modifications can be easily made with a direct control on the output returned by the system. But there are some issues that still remain beyond the scope of chronos system. The issues such as normalization problem is raised by reported speech fragments, anaphoric expressions raise a detection problem, and ambiguous expressions, apparent dates raise the problem of spurious tagging

4.3 TIMEML

TIMEML [10][11] annotation scheme identifies timexes and also events in the text document. TIMEML consider tensed verbs, stative adjectives, event nominal's as classes of event expressions and relate time expressions using temporal connectives and temporal propositions. These relations are termed as signals. The representation in this markup language has a closer affinity to natural language rather than to inference formalism. The markup language has the mechanism for systematic anchoring of events for time expressions, ordering of events, and also interpretation of underspecified temporal expressions.

TimeML [15] uses 14 temporal relations in the TLINK relTypes. The relTypes are *TRels*= {SIMULTANEOUS, IBEFORE, BEFORE, BEGINS, ENDS, INCLUDES}. These relations are similar to qualitative interval algebra (IA). Inorder to have a non hierarchical classification, SIMULTANEOUS and IDENTITY are collapsed, since IDENTITY is a subtype of SIMULTANEOUS. Similarly DURING and IS_INCLUDED are collapsed, since DURING is a subtype of IS_INCLUDED. IBEFORE (immediately before) corresponds to MEETS in Allen's Interval calculus. Allen's OVERLAP relation is not represented in TimeML. By using all these considerations TLINK relations collapses to a disjunctive classification of six temporal relations. Six relations and their inverses map one-to-one to 12 of Allen's 13 basic relations

Machine learning rules are applied to interpret the temporal expressions [16]. These rules solves generic/specific use of today (e.g., "today", now" etc.) and direction problems. The rules for disambiguating "today" problem check for features such Poss (whether "today" has a possessive inflection), Qcontext (whether "today" is inside a quotation), presence of words such as "said", "will", "even", "most", "some", "year", "day of the week" in the same sentence, FW ("today" is the first word of the sentence), POS1 (parts-of-speech of word before "today"), PO2 (parts-of-speech of word after "today"). "Today" is classified as generic if any one of poss, most and FW is true and is referred as specific if "day of the week" is mentioned anywhere in this sentence.

The direction problem is solved by comparing the tense of the neighboring verbs. When it is the past tense the direction is before the reference time, while the future tense and the present tense indicate after and at reference time respectively. The features such as day names, month, and years is extracted to timestamp for fixing the reference time.

Logical inferences in TIMEML is not trivial especially to recover information which has not been mentioned explicitly in the document, therefore it requires mechanism to leverage machine learning for unlabeled data. It is expressive to record about event structure information with exceptions of quantification in natural language text. The designers of TIMEML have struck between expressive complexity and computational simplicity.

5. DOMAIN SPECIFIC MODELS

Temporal information extraction in domain specific models [7] is based on corpus built using human annotation. These models are very specific and are restricted to a particular domain and often criticize to be based on manually built corpus.

6. COMPARISION

Table 1: Comparision of constraint based, semantic based and domain specific models

	Constraint based	**Semantic based**	**Domain specific**
Expressibility	Less, since all the temporal information is represented as temporal calendric constraints	High, because identification of temporal information is based on annotation schemes	Less,It can identify only specific time expression
Scope of semantics	Temporal information such as (negation) cannot be identified	Negation can be identified	Does not exist
Ambiguity problem	Distinguishing the ambiguity between deitic and temporal time expressions is left incomplete	It can address many of the ambiguity problems among time expressions by interpreting the values of attributes in the tag.	Does not occur
Tractability	Subset of relations which are referred as tractable	Well defined procedure for solving computational problems need	No tractability

	classes make the computation easier	to be established and hence tractability problem is still open	
Error analysis	Error rate is less both in recognition and normalization of temporal expressions	Error rate is high in recognition and normalization, when machine learned classifiers are applied	Error rate almost tends to zero as the expression are fixed.

7. CONCLUSION

The present work is a survey of popular models used for temporal extraction and interpretation. On comparing constraint based, and semantic based models, it has been observed that while constraint based models provide a great flexibility to the user in terms of expressibility, there is no way to specify semantic scope and thus problems such as negation cannot be represented. Semantic models address the ambiguity problems and also can handle shift in temporal focus with a dynamic changes in the natural language text. However these methods are not sound and complete for reasoning with temporal information. Domain specific models attempt a tradeoff between expressibility and tractability, however the range of NLP problems solved and represented in this models are limited and biased. The automated reasoning methods in constraint based models can be utilized by choosing appropriate subset of semantic models to achieve decidability without sacrificing the expressiveness.

8. REFERENCES

[1] Allen J.F, 1983, "Maintaining knowledge about temporal intervals". Communications of the ACM 26 , pp-832–843.

[2] Arun K. Pujari, G. Vijaya Kumari, Abdul Sattar, "INDU: 1999, An Interval and Duration Network", in Proceedings of the 12th Australian Joint Conference on Artificial Intelligence: Advanced Topics in Artificial Intelligence Vol 1714, pp 291-303.

[3] Benjamin Han, 2003 "Time calculus for natural language - tagging guidelines." Unpublished draft, Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA..

[4] Bernhard Nebel, Hans-Jürgen Bürckert, 1995, "Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra " Published in the Journal of ACM 42(1): pg:43-66.

[5] David Ahu, Sisay Fissaha and Maarten de Rijke, 2005, "Towards Tasks based Temporal Extraction and recognition", In the proceedings of Annotating, Extracting and Reasoning about Time and Events, International Seminar, Dagstuhl Castle, Germany, April 10-15..

[6] Ferro L., Mani I., Sundheim B., and Wilson G. 2000, TIDES Temporal Annotation Guidelines. Draft Version 1.0. MITRE Technical Report MTR 00W0000094

[7] Frank Schilder, 2007, "Event Extraction and Temporal Reasoning in Legal Documents" , In the proceedings of Annotating, Extracting and Reasoning about Time, LNAI-Volume 4795/2007, pp:59-71..

[8] G Vijaya kumari, Arun K pujari, 2002, "Maximality of pre-convex class in INDU", KBCS

[9] Han B. and Kohlhase M. 2003, "A time calculus for natural language". In *Proceedings of the 4th Workshop on Inference in Computational Semantics* (Nancy, France).

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

[11] Inderjeet Mani and Ben Wellner. 2006. A Pilot Study on Acquiring Metric and Temporal Constraints for Events. In Proceedings of Annotating and reasoning about Time and Events (ARTE). Sydney, Australia.

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

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

[14] Negri M. and L. Marseglia,. 2005, Recognition and Normalization of Time Expressions: ITC-irst at TERN 2004. Technical Report WP3.7, Information Society Technologies

[15] Peter van beek, Robin Cohen, 1990, "Exact and Approximate reasoning about Temporal, Relations", Journal of Computational Intelligence, Vol 6, pp 132-144.

[16] Roser Saurí and Marc Verhagen..2005, Temporal Information in Intensional Contexts. Bunt, *IWCS-6. Sixth International Workshop on Computational Semantics*: pp 404-406..