Temporal and Event Information in Natural Language Text

James Pustejovsky, Robert Knippen, Jessica Littman, Roser Saurí
Brandeis University

Abstract. In this paper, we discuss the role that temporal information plays in natural language text, specifically in the context of question answering systems. We define a descriptive framework with which we can examine the temporally sensitive aspects of natural language queries. We then investigate broadly what properties a general specification language would need, in order to mark up temporal and event information in text. We present a language, TimeML, which attempts to capture the richness of temporal and event related information in language, while demonstrating how it can play an important part in the development of more robust question answering systems.

Keywords: question answering, temporal ordering, annotation, events, modality, temporal expressions

Keywords: question answering, temporal ordering, annotation, events, modality, temporal expressions

Table of Contents

1. Introduction

The automatic recognition of temporal and event expressions in natural language text has recently become an area of intense research in computational linguistics and Artificial Intelligence. The importance of temporal awareness to question answering systems has become more obvious as current systems strive to move beyond keyword and simple named entity extraction. Named entity recognition (Chinchor et al, 1999) has moved the fields of information retrieval and information exploitation closer to access by content, by allowing some identification of names, locations, and products in texts. Beyond such metadata tags, however, there is only a limited ability to mark up text for real content. One major problem that has not been solved is the recognition of events and their temporal anchorings.

Newspaper articles describe the world around us by talking about people and the events and states of affairs they participate in. As it happens, however, much of the temporal information in a report or narrative is left implicit in the text. The exact temporal locations of events is rarely explicit and many temporal expressions are vague at best. A crucial first step in the automatic extraction of information from such texts, for use in applications such as automatic question answering or summarization, is the capacity to identify what events are being described and to make explicit when these events occurred.

While questions such as the following can be easily answered by human beings after reading the appropriate newspaper article, such capabilities go beyond any current automatic system:

(1) a. Is Schröder currently German chancellor?

b. What happened in French politics last week?

c. When was the merger between Deutsche Bank and Dresdner Bank?

The recognition of temporal "keywords" (e.g., *currently*, *last week*) is clearly a prerequisite for understanding and answering these questions. In addition, further temporal knowledge needs to be represented and further temporal inferences need to be drawn. First, temporal aspects of the properties of entities (i.e., property of being German chancellor) must be adequately represented. Second, the extraction of event descriptions with their time stamps has to be carried out. The knowledge of certain temporal features of events (i.e. typical duration of an event) seems also to be crucial for the correct understanding of text. Finally, the veridicality of events has to be checked as well (i.e., actual vs.

intended events). As can be seen from these three example questions, building an automatic system that can extract and reason with temporal and event information bringsup new multifaceted research issues. First we require an expressive language in which the kind of event and time information we are concerned with can be made explicit.

2. Temporal Information in Questions

Natural language questions express possible queries that a QA system must answer. In many of them, temporal information is a basic component of knowledge, and needs to be handled for an acceptable degree of performance to be attained. Take as example the following interrogative sentences, which are extracted from (or based on other questions in) the Excite question log.

(2) a. When did Yale first admit women?

b. How long does it take to climb Everest?

Both examples in (2) are looking for a temporal value associated with the expressed event: a date in the case of (2a), a duration in (2b).[1] Answering (2a) does not require a very powerful reasoning engine. Assuming that the needed information is contained in our knowledge base, the answer will be the specific date associated with the event denoted by the query. For unstructured textual knowledge bases, it will suffice to have a system capable of identifying events and temporal expressions in a text and anchoring the events in the timeline. The first task can be performed reasonably well by a chunker informed with lexical information and constrained to a limited set of structures ((Mani and Wilson, 2000), (Schilder and Habel, 2001), (Pustejovsky et al., 2002)). Similarly, some time-stamping relations can be extracted from the parse tree, as is done in previous work ((Filatova and Hovy, 2001), (Mani et al., 2003)). . However, queries like (2b) demand a higher temporal reasoning capability. The eventive predicate (*climbing Everest*) does not refer to a unique event in the knowledge base, and therefore the felicitous answer will need to be calculated over a set of temporal expressions presumably associated with that expression. A similar issue is at play in the following pair of examples:

(3) a. Who was the American ambassador to Japan before Walter Mondale?

[1] Here and throughout the article we use the term 'event' to refer to the ontologically broader notion of eventuality. Our events will therefore include states as well as dynamic events.

b. What is the name of the teacher that went to jail after getting pregnant by a student?

Answering the examples in (3) involves dealing with temporal relations such as *before, after*, and *during*. The reasoning system does not need to be very sophisticated for questions like (3a) which, similar to (2a), focusses on very unique events. The first time Yale admitted women happened presumably only once in our timeline. Similarly, there is only a limited number of states of somebody being the American ambassador to Japan, and all of them are reasonably well anchored in time. From a structured knowledge base perspective, the answer to (3a) can be provided on the basis of ordered lists of domain-prominent state-of-affairs in the world. From less structured sources, the processing tools mentioned above for question (2a) would help in time-stamping and ordering the set of states of being an American ambassador to Japan. However, answering queries like (3b) demands the temporal ordering of events that may not be as well temporally delimited as those in (2a) or (3a), and for which no precise time-stamping may exist. Consider now the examples below:

(4) a. When is the Monsoon season in Southeast India?

b. Between October and December.

The question in (4a) asks for the temporal value related to a temporal expression (*Monsoon season*, which is of the same nature as *Ramadan, Christmas*, or *Passover*). Assuming that our knowledge base contains the necessary information to answer it, we can expect Monsoon season to be temporally related to a bounded period of time (such as `between(Monsoon,October,December)`), so that answer (4b) is returned.

(5) a. When does the Monsoon season begin in Southeast India?

b. In October.

(6) a. How long does the Monsoon season last in Southeast India?

b. For three months.

Additional mechanisms will be needed, however, in examples (5-6), both for interpreting the queries (so that the references to only part of the Monsoon period in (5a) and to its duration in (6b) are identified), and in the answering process in order to compute the appropriate information from the statment in our knowledge base. Whatever strategy a QA system applies, it is clear that answering those questions requires controlling information along the temporal axis. Other examples illustrating the fundamental character of temporal information in queries are the following:

(7) a. Who won the Nobel prize this year?

b. Who is the President of Argentina?

As in (3), neither of the two questions above inquire about a temporal expression or relation, but about individuals. Yet, answering them involves locating the events they refer to in the time axis. The structure of (7a) ressembles that of (2a) in that they both relate an event (*winning the Nobel prize* and *admitting women to Yale*, respectively) to a temporal expression (in (7a), *this year*, and in (2a) the value that will be returned as answer to the question). Furthermore, temporal reasoning is also essential in queries with no overt or queried temporal values. Question (7b), for instance, may not receive an adequate answer if no reference to the temporal axis is made.

Examples (2-7) illustrate the extent to which temporal information is pervasive throughout questions, thereby demanding systems capable of representing and reasoning with temporal knowledge. This is the case for many of the different types of queries in the typologies used by current QA systems (e.g., (Abney et al., 2000), (Hovy et al., 2002)). We will now look into different kinds of questions and analyze the relevant components that must be identified in order to answer the question felicitously. The first task to address here is the identification of temporally relevant queries since, as shown in (3) and (7), it is not necessary for them to be introduced by a *wh*-word referring to a temporal index, nor to contain an overt temporal expression. Still, a generalization applies to all sentences in (2-7). They all involve at least one temporal relation of any of the following sorts:

- Between an event E and a temporal reference T . The temporal reference T can be explicit (8a), implied by the *wh*-word (2), or contextually implied as in (7b);

- Between an event E and another event E' (8b);

- Between two temporal values, T and T' (8c).

(8) a. How many servings of Coca-Cola were consumed in 1994?

b. How many Iraqi civilians were killed during the attack on Falluja?

c. When is Chanukah?

Taking these three relation types as defining the nature of temporally relevant queries, we now have a better view on the kinds of potential questions that are involved here.

A first, unequivocal subclass of temporal queries consists of queries that look for a temporal value as its felicitous answer, be it an index to a calendar date (9a), an index to a time of the day (9b), a duration (9c), or a set (9d). For expository purposes, we identify this required temporal value as **qT**,[2] and the queries featuring them, as **qT-queries**.

(9) a. When is the next full moon?

b. What time is The Daily Show?

c. How long does it take to climb Everest?

d. How many days is the temperature below 32 °F in Barrow, Alaska?

Linguistically, queries of this sort are distinguished by a specific set of *wh*-phrases. Queries aiming at durations are generally introduced by (10a), whereas those looking for a set, by expressions like (10b). On the other hand, queries pointing to temporal indices (calendar dates and time of the day) are introduced by phrases like (10c), where N_t is any temporal denoting noun (of the sort *hour, day, month, century, Wednesday, January*, etc.), and NP_t is an NP headed by an N_t.

(10) a. *how long*

b. *how often*
how many times

c. *what* + N_t
what is + NP_t
on what + N_t

Of course, the class of queries described here is also, and very commonly, introduced by the very distinctive temporally-selecting *wh*-word

[2] qT and the other q-terms we will introduce in the remainder of the article (qE, qI, qV, and qR) correspond broadly speaking to Qtargets in the QA literature (cf. (Hovy et al., 2002)).

when, which is not restricted to any of the temporal value types distinguished in (9).

qT-queries can be nicely classified according to the kind of temporal relations they convey. Some of them involve a relation qT-T, between the queried temporal value qT and another temporal value T (11a). Some others hold a qT-E relation between qT and an event E (11b). Others present two relations: a first one between qT and an event E, which in turn is related to a time value T (as in 11c). Using the same notation as for the previous two relations, this case can be represented as qT-(E-T), but for simplicity's sake, they will be represented as qT-ET. Queries of the form qT-TT are also possible (11d).[3]

(11) a. **qT-T:** [$_{qT}$ When] is [$_T$ the first day of winter 1999]?
b. **qT-E:** [$_{qT}$ What year] was the toilet [$_E$ invented]?
c. **qT-ET:** [$_{qT}$ What is the last day] [$_E$ to contribute to a Roth IRA] for [$_T$ 1999]?
d. **qT-TT:** [$_{qT}$ When] was [$_T$ Ramadan] [$_T$ this year]?

Note that because of the polymorphic nature of the *wh*-word *when*, qT-queries introduced by this particle can actually receive as answer a reference to the temporal relation that E or T holds with a second event, E' (12c):

(12) a. When did the embargo on Iraq begin?

b. In mid_September 1988.

c. Before the Kuwait crisis.

In addition to qT-queries, other temporally relevant queries are **qE-queries**. They look for events as their appropriate answer type (**qEs**, in our terminology). All the cases identified as qE-queries are introduced

[3] Note, however, that they are restricted to two very specific cases:

(i) [$_{qT}$ When] was [$_T$ Ramadan] [$_T$ this year]?

(ii) a. [$_{qT}$ What day] is [$_T$ December 18th] [$_T$ this year]?
b. [$_{qT}$ What week] is [$_T$ Feb 2nd] [$_T$ this year]?

qT-TT queries like the one in (i) can only be constructed with event based temporal expressions as the T to which qT is related (*Ramadan* in the current case). Temporal expressions of this sort are: religious festivities (*Chanukah* or *Easter*), other local-based festivities (*Carnaval*, *Les Santes*), weather-based seasons (*Monsoon*, *hurricane season*), etc. On the other hand, qT-TT queries conforming to the model in (ii) are restricted to having a calendar date as the T to which the qT is related, and accept only days of the week, or week number, as possible values for the qT, respectively.

by the expression *what happened* and, interestingly, the set of temporal relations they can convey is equivalent to the one shown for qT-queries. In other words, qE-queries also hold relations between the qE element and a temporal reference T (13a), another event E (13b), an event E that is anchored to a temporal value T (13c), or a time reference T that is anchored to another temporal value T' (13d).

(13) a. **qE-T:** $[_{qE}$ What happened$]$ in Czechoslovakia $[_{T}$ in 1968$]$?
b. **qE-E:** $[_{qE}$ What happened$]$ in Vietnam after $[_{E}$ the war$]$?
c. **qE-ET:** $[_{qE}$ What happened$]$ during $[_{T}$ yesterday$]$'s $[_{E}$ strike$]$?
d. **qE-TT:** $[_{qE}$ What happened$]$ during$[_{T}$ Ramadan$]$ $[_{T}$ this year$]$?

Other temporally relevant queries are those aiming at answers typed as individuals or values (**qI** and **qV**, respectively). qI and qV are in fact arguments of an eventive relation expressed or implicated in the query. **qI-queries** are exemplified in (14), **qV-queries**, in (15).

(14) a. Who was the ruler of Egypt when Jesus Christ was born?

b. What president had two Vice Presidents die while in office?

(15) a. How many Iraqi civilians have been killed in the last year?

b. How old was Che Guevara when he was killed?

qI-queries are introduced by *wh*-phrases like those in (16), where N_i is any individual denoting noun, and NP_i is an NP headed by an N_i. Similarly, qV-queries are distinguishable by *wh*-phrases of any of the patterns in (17):

(16) *who*
what + N_i
what is/was + NP_i

(17) *how many/much* + $N_{non-temporal}$
how + Adj_{scalar}
what + $N_{dimension}$
what is/was + $NP_{dimension}$

As with qT- and qE-queries, qI- and qV-queries reproduce the same correlations between the q-term and the element it is temporally related to. This is actually predictable from the fact that q-terms in qI- and qV-queries are arguments of an event-denoting predicate, which is, strictly speaking, the element participating in the temporal relation. The query types for qI- and qV-queries (corresponding to those in (11) and (13) for qT- and qE-queries) are here characterized as expressions of the form $\mathbf{q}_\chi\mathbf{E}\text{-}\Upsilon$, where χ is a variable over types I (for individuals) and V (for values), q_χE is a q-term of type I or V which refers to an argument of event E, and Υ is a temporal entity (either a time value T or an event E') to which E is temporally related.

(18) a. $\mathbf{q}_I\mathbf{E}$**-T:** $[_{q_I E}$ $[_{q_I}$ Who] was born] on $[_T$ December 18th]?
b. $\mathbf{q}_I\mathbf{E}$**-E:** $[_{q_I E}$ $[_{q_I}$ Who] was the ruler of Egypt] when Jesus Christ $[_E$ was born]?
c. $\mathbf{q}_I\mathbf{E}$**-ET:** $[_{q_I E}$ $[_{q_I}$ Who] was killed] during $[_T$ yesterday]'s $[_E$ strike]?
d. $\mathbf{q}_I\mathbf{E}$**-TT:** $[_{q_I E}$ $[_{q_I}$ Where] did G.W. Bush travel] for $[_T$ Thanks- giving] $[_T$ last year]?

(19) a. $\mathbf{q}_V\mathbf{E}$**-T:** $[_{q_V E}$ $[_{q_V}$ What] was the lowest temperature] $[_T$ last winter]?
b. $\mathbf{q}_V\mathbf{E}$**-E:** $[_{q_V E}$ $[_{q_V}$ How old] was Che Guevara] when he $[_E$ was killed]?
c. $\mathbf{q}_V\mathbf{E}$**-ET:** $[_{q_V E}$ $[_{q_V}$ How many students] were killed] during $[_T$ yesterday]'s $[_E$ strike]?
d. $\mathbf{q}_V\mathbf{E}$**-TT:** $[_{q_V E}$ $[_{q_V}$ How many turkeys] were eaten] for $[_T$ Thanksgiving] $[_T$ last year]?

Contrary to the other temporal query classes seen so far (qT- and qE-queries), qI- and qV-queries can also be temporally relevant and yet not express a temporal relation overtly, as illustrated in examples (7b) above and (20):

(20) a. What company is ranked number 1 on the Fortune 500 list of companies?
b. What is the population of Iraq?

Examples above are queries looking for an individual I or a value V that is the argument of an event E_n, computed from a set of temporally-ordered events Σ. All elements in Σ have equivalent intension, but receive a different extension depending on the temporal index they are related to. Thus, the answer in (20a) will refer to a different company depending on the year the state of *being ranked number 1 on the Fortune 500 list of companies* is anchored to. Similarly, the value indicating the population in Iraq will vary from day to day.

We will represent these queries as $\mathbf{q}_\chi\mathbf{E}_n$**-(T)**, where χ is again a variable over types I and V, E_n refers to an event $\mathrm{E} \in \Sigma$ (the set of temporally ordered events), $\mathrm{q}_\chi\mathrm{E}_n$ denotes a q-term of type I or V which is an argument of E_n, and T is an implicit temporal index.[4] Two distinctive strategies for locating event E_n within Σ can be distinguished. On the one hand, there are cases like those in (21) in which E_n is calculated on the basis of an ordinal term (*first,third*) pre-modifying the event-denoting expression. Knowing the temporal anchoring of E_n to T is therefore not strictly necessary in examples like those above. T will correspond to the temporal index of E_n, $\mathrm{T} = \mathrm{T}_{E_n}$.

(21) $\mathbf{q}_\chi\mathbf{E}_n$**-(T)**, where $\mathrm{T} = \mathrm{T}_{E_n}$:

a. $\mathbf{q}_I$**E-(T):** Who won the *first* Rose Bowl game?
b. $\mathbf{q}_V$**E-(T):** What was the score of the *third* Rose Bowl game?

This is however not the case with queries in (22-24), where the anchoring to an absolute temporal index (by default a present reference, T_{now}) is crucial. Differences in the linguistic encoding of the information allow for grouping those queries into several subclasses, which, correspondingly, requires different reasoning schemes.

(22) $\mathbf{q}_\chi\mathbf{E}_n$**-(T)**, where $\mathrm{T}_{E_n} \prec \mathrm{T}$, or $\mathrm{T}_{E_n} \succ \mathrm{T}$, or $\mathrm{T}_{E_n} = \mathrm{T}$:

a. $\mathbf{q}_I$**E-(T):** Who was the *previous* President in Catalonia?
b. $\mathbf{q}_V$**E-(T):** What was the *previous* lowest temperature registered?

(23) $\mathbf{q}_\chi\mathbf{E}_n$**-($\mathbf{T}_{now}$)**, where $\mathrm{T}_{E_n} = \mathrm{T}_{now}$:

a. $\mathbf{q}_I$**E-(T):** Who is doing the body count in Iraq?
b. $\mathbf{q}_V$**E-(T):** How old is Michael Jackson?

(24) $\mathbf{q}_\chi\mathbf{E}_n$**-($\mathbf{T}_{now}$)**, where $\mathrm{T}_{E_n} = \mathrm{T}_{now}$:

a. $\mathbf{q}_I$**E-(T):** Who is the President of Venezuela?
b. $\mathbf{q}_V$**E-(T):** What is the temperature in Ellicotville, NY?

[4] As the notation suggests, these queries correspond to subtypes of q_IE-T and q_VE-T, exemplified in (18a) and (19a), respectively.

In some cases (such as 22), a sequencing modifier (*previous, next, current*) is employed to signal the temporal relation between E_n and the implicit temporal index T. Depending on the sequencing term, the relation between the two entities will be $\prec$, $\succ$, or $=$. A reference to a temporal index is also needed in queries grouped under (23), which are different from the previous cases in that E_n is not explicitely temporally ordered. In this case, the temporal relation assumed by default is $\mathrm{T}_{E_n} = \mathrm{T}_{now}$. Queries exemplified by (24) are very similar to those in (23), the main difference being that in these the predicative force is carried by either an agentive nominal (*president, landlord, passenger*) or a measure-denoting noun (*population, height, temperature*).

At a higher order of complexity, there is the class of queries looking for the value of the temporal relation itself, identified here as **qR-queries**. Again, they can be subclassified depending on the types of the entities involved in that relation:[5]

(25) a. **qR(T-T):** Is [$_T$ Ramadan] [$_{qR}$ before or after] [$_T$ Christmas]?
b. **qR(T-E):** Was [$_T$ Thanksgiving] [$_{qR}$ before or after] [$_E$ the 9/11 Comission report]?
c. **qR(T-ET):** Was [$_T$ Thanksgiving] [$_{qR}$ before or after] [$_E$ the 9/11 Comission meeting] in [$_T$ November?
d. **qR(T-TT):** Is [$_T$ Easter] [$_{qR}$ before, during or after] [$_T$ Passover], [$_T$ this year]?

(26) a. **qR(E-T):** Was [$_E$ the attack on Falluja] [$_{qR}$ after or during] [$_T$ Ramadan]?
b. **qR(E-E):** Did [$_E$ J. Kerry concede] [$_{qR}$ before or after] [$_E$ finishing the ballot counting in Ohio]?
c. **qR(E-ET):** Was [$_E$ the graduation ceremony] [$_{qR}$ before or after] [$_E$ the soccer finals], [$_T$ last year?
d. **qR(E-TT):** Was [$_E$ the graduation ceremony] [$_{qR}$ before or during] [$_T$ May] [$_T$ last year]?

Finally, there are also **Yes/No queries** inquiring about the truth value of a temporal relations made explicit in the text:

(27) a. **q(T-T):** Is [$_T$ Lent] before [$_T$ Carnaval]?
b. **q(T-E):** Was it [$_T$ Ramadan] during [$_E$ the attack on Falluja]?
c. **q(T-ET):** Was it [$_T$ night time] when [$_E$ the suspect arrived

[5] Note that the reasoning capability needed for answering queries like the following is very similar to that required for obtaining the alternative answers of qT-Υ queries in (12), where Υ denotes an entity of temporal nature (T or E).

in Boston] on [$_T$ January 8th, 2001]?
d. **q(T-TT):** Was [$_T$ Thanksgiving] during [$_T$ Ramadan] in [$_T$ 2003]?

(28) a. **q(E-T):** Are poinsettias [$_E$ popular in Australia] during [$_T$ Christmas]?
b. **q(E-E):** Did Putin [$_E$ lift the embargo] on Iraqi arms sales before [$_E$ the end of the war]?
c. **q(E-ET):** Were WMD [$_E$ found] before [$_E$ the attack on Iraq] in [$_T$ 2003]?
d. **q(E-TT):** Was there any [$_E$ combat] during [$_T$ Ramadan] [$_T$ last year]?

Temporal information is therefore an important component of different kinds of questions in natural language. In some cases, it is present even if there is no explicit temporal reference, be it a *wh*-expression or an overt temporal expression. A QA system must be sensitive to the different ways temporal relations are conveyed in natural language queries, as well as capable of controlling the information at the temporal axis.

3. Representation of Temporal Information for QA Systems

The previous section made clear how important it is that QA systems are sensitive to temporal information of various sorts. In this section, we consider more specifically the kinds of temporal information that might be needed for answering questions and how this information might be represented for use by a QA system.

We assume that the construction of a knowledge base for QA will involve marking up a document collection with some annotation language, so the question we address here is what such a markup language has to be like in order to use it to annotate documents for temporally sensitive question answering. There are clearly types of information that such a language must be able to encode if it is going to be at all useful for retrieving and inferring temporal information from texts. For example, it should be clear from Section 2 that to be useful for retrieving answers to temporally-sensitive questions, a knowledge representation language must have some way to annotate the way natural languages refer to events, times, and temporal relations. We can see that any temporally-aware QA system must have the ability to anchor events in time and order them. By *event anchoring* we mean placing a given event on a timeline. By *event ordering*, we mean establishing the relative position of two events in time. To make either of these tasks possible, a language must have some way of uniquely identifying events and times, as well as a way to express relations between these two. The details of how these three primitives are expressed in English, as well as their conceptual background, have been discussed elsewhere (e.g., (Setzer, 2001)). Here we will be concerned with those those features of events, times, and relations which must be encoded in order to answer questions effectively.

The information extraction component of QA is also significant, so we will also have to consider the sorts of time-related information needed by tools and algorithms which automatically extract temporal information. It is important to us that any annotation scheme is equally useful as a language for marking up corpora which can later be used to train and evaluate temporal information extraction algorithms. Information about the tense and aspect of finite verbs, for instance, may not be directly useable for answering questions, but as an important way that natural languages express time, it may be useful to algorithms which determine the anchoring and ordering of events in texts. Further, given the range of linguistic mechanisms involved in expressing temporal information, a language which allows an incremental, layered approach to information extraction will be valued highly.

3.1. Retrieving information from texts

3.1.1. *Events*

Events, as well as the kinds of states which change and thus might need to be located in time (we will refer to these as events here), are referred to by finite clauses, nonfinite clauses, nominalizations, event-referring nouns, adjectives, and even some kinds of adverbial clauses, as seen in the following examples.

(29) a. When President Leonid Kravchuk was *elected* by the Ukrainian Parliament in 1990, he vowed to seek Ukrainian sovereignty.

b. In July 1994, Ukraine again held free and fair *elections*.

c. *Vowing* to seek Ukrainian sovereignty, Kravchuk ...

d. While *in office*, Kravchuk was always an advocate for ...

As mentioned above, a language for representing temporal information in texts must have a way of identifying events so they can be anchored and ordered. Just as important for retrieval in QA is some way to indicate whether an event-referring clause includes a negation of the relevant event, as seen in the following.

(30) a. When it became clear that controllers could not *contact* the plane, ...

b. No one *reached* the site in time.

English has a wide range of mechanisms for expressing what might be referred to as a 'negative' event. It might not be possible or necessary to design an annotation system which indicates how the negation is expressed in every case. However, in order to determine whether one of the statements in (30) constitutes an answer to some question, it must have some way to record the fact that the relevant event is negated in each case.

Just as important, though perhaps more difficult for a retrieval system to deal with, is the fact that events are often expressed with various types of modality, as seen below.

(31) a. The United States **may** *extend* its naval quarantine to Jordan.

b. Some assets **might** be *sold* to service the debt.

c. The deal **must** *give* inspectors unrestricted access.

d. Sununu has plenty of support and **should** be *appointed* ...

Epistemic modality, seen in examples (31a) and (31b), has to do with aspects of events such as necessity and possibility. Deontic modality, seen in examples (31c) and (31d), has to do with aspects of events such as obligations agents might have with respect to them, or the permissibility of events. In general, such modality is expressed in English with a modal verb. The modality expressed in each case clearly has implications for the suitability of statements as answers to questions. With no more information than (31a), for example, a question answering system should not treat the event referred to by *extend* as if it really occurred. One can even imagine domains in which questions refer directly to modalities such as permissibility (*Can a US citizen visit Cuba before 2005?*). Thus, a retrieval system should at the very least record the modality of the events in statements like the above.

In section 4, we discuss a way to annotate events that allows information about negation and modality to be represented.

One factor complicating the markup of events in texts is the fact that not every unique event referred to can be associated with a text extent. That is, some text extents refer to multiple events, by quantifying over events as in (32a), or by the use of various kinds of syntactic ellipsis, as in (32b).

(32) a. James *taught 3 times* on Tuesday. The first time was at 8am.

b. Marty *taught* on Friday, but James didn't.

Because the different events referred to by a single text extent may have different negation and modality properties, as well as different sorts of relations to times and other events, it is important for any annotation language to have a way to reify the multiple events referred to in a single text extent. In (32b), two teaching events are referred to, but one is negated. In (32a), there are 3 events of teaching referred to, and they must be represented seperately to capture the different times expressed for each one. In section 4, we discuss a method for associating multiple events with a single text extent.

3.1.2. *Times*

The main reason for the knowledge base of a QA system to concern itself with time expressions is in order to be able to anchor events to times. It may sometimes be useful for efficient retrieval to order times with respect to each other, but the main concern will always be trying to place events on a timeline. Times are usually expressed in English by adverbial or prepositional phrases, as shown in the following:

(33) a. on Thursday

b. November 15, 2004

c. Thursday evening

d. in the late 80s

e. later this afternoon

f. yesterday

In order to anchor events to times on a timeline so questions can be answered, it is necessary to normalize time expressions to a representation that can be mapped to a timeline. Such a normalization simplifies by conflating the different ways English has for referring to the same time (e.g., *11/15/04, November 15, 2004, the 15th of November in 2004*, etc.). It also resolves any indexical component there might be to a time. Many time expressions refer to a point in time via some indexical anchor, as seen in the following:

(34) a. today

b. next Friday

c. last week

d. in October

All of these expressions refer to a time, but they do not by themselves fully specify that time. They refer via reference to the moment of utterance–in the case of texts, the document creation time. One has to know the time of utterance in order to retrieve the time referred to and normalize them to some machine-readable form. ISO 8601 provides a useful standard for the purpose of normalizing times. However, English has numerous ways to express what might be called 'indeterminate' times, which cannot be determinately linked to a timeline. Some examples follow.

(35) a. in the Fall of this year

b. recently

c. yesterday morning

Such times cannot be interpreted directly as parts of a timeline, because their begin and end points are more or less vague. Nevertheless, they can be ordered with respect to most points on a timeline, and so a knowledge representation system for QA must have some way of normalizing them. In Section 4 we discuss a set of indicators that are useful for normalizing many such expressions.

The time expressions mentioned so far refer, with greater or lesser granularity and with greater or lesser precision, to coherent 'chunks' of the timeline. They provide a means for directly associating particular events with particular parts of the timeline, and are thus of primary importance in QA applications. English contains two more kinds of time expressions which involve slightly more complex means of anchoring events to times. The first of these two types is the duration. Durations refer not to parts of the timeline, but to quantities of time.

(36) a. after *three weeks*

b. a *day*

c. for *three hours*

d. a *two-hour* flight

The time expressions in these examples simply indicate the duration of events. They thus might be significant to making inferences about how events are ordered with respect to each other, or making inferences about the location of a particular event on the timeline. These topics will be discussed below. One also cannot rule out the possibility that a QA system might be presented with a query about the duration of a particular event. The representation of such expressions must normalize these periods of time so inferences can be drawn. Other durations are part of a more complex system for indicating the time of an event.

(37) two weeks from *today*

We call this latter type of duration *anchored durations*, because they express the time of an event by making explicit the duration of time between the event and a time. In fact, they can be said to be part of a compositional time expression. For example, in (37), the duration *two weeks* is anchored to the time expression *today.* Thus, in combination with the temporal preposition *from* and the time expressed by *today*, it refers to a time two weeks after the document creation time. In order to effectively answer questions about the event referred to, a QA retrieval system should have some access to the time referred to. It could be retrieved via a calculation from the anchor, the duration, and the nature of the relation indicated by *from*, or this information could be calculated by an information extraction system and stored as part of the annotation of this statement. The language for representing time in QA systems should allow for either possibility.

Note that durations can also be used to anchor events to other events.

(38) Three weeks before the *invasion*, most stockpiles were *destroyed.*

The time expression here does not refer to parts of the timeline, but indicates distance along it–the amount of time that separates the italicized events. As such, it does not directly anchor events to times, but may allow the time for an event to be inferred. Like durations anchored to times, the amount of time they indicate should be represented in any temporal language for question answering.

The final type of time expression to consider is exemplified below.

(39) a. every Thursday

b. two weekends per month

These time expressions indicate what are referred to as sets of times. They refer neither to coherent chunks of the timeline, nor to distances along it, but to, roughly speaking, groups of distinct pieces of the timeline. They are used to place recurring events on the timeline, particularly when the recurrence is regular. While corpus study reveals that such time expressions are not common in English texts, they are one way English allows events to be associated with the timeline, and a language for representing the temporal aspects of English texts should have some way of normalizing them. A way to do so will be discussed in Section 4.

3.1.3. *Temporal Relations*

In order to perform the fundamental tasks of anchoring and ordering events, the last major building block required of a temporal annotation language is the ability to represent temporal relations. The language must have some way to characterize the relationship between events and times. English and other natural languages do not usually express the interval which a given event takes on the timeline directly, in terms of its specific endpoints. Instead, they use a range of strategies to indicate a relation between a given event and other times and events in the text. In most cases, the result is that the interval which a given event takes on the timeline is expressed only partially.

(40) A Brooklyn woman who was watching her clothes dry in a laundromat was *killed* on <u>Thursday evening</u> ...

In this example, a temporal relation which we might express as *is included* is predicated between a time and the event referred to. It expresses that the the interval in which the event *killed* occurs is included in the time *Thursday evening.* This is signalled by the presence of the preposition *on.* This information is partial in that the precise begin and end points are not specified for the event. The reader only knows that it occurred somewhere within a particular range on the timeline. If specific end and begin points for events were always expressed, it would be possible for a knowledge base to directly represent them as part of the event. However, a relation such as 'is included' anchors an event to a time by specifying a pair of ordering constraints on the endpoints of the event; its begin point is after the time referred to by *Thursday evening* begins and its end point is before *Thursday evening* ends. Such a complex pattern is best expressed by a relation which can easily be interpreted by machine. The need for relations is seen even more clearly in examples like the following.

(41) a. Not that long ago, before the Chinese *takeover*, real estate prices in Hong Kong *hit* a record high.

b. We were *eating* dinner when the wall *fell* on us.

c. After John *left*, I *realized* he still had my pen.

In these examples, the only information that is expressed about the temporal extent of the events are ordering constraints on the begin and end points of the events. (41a) is an example of a *before* relation, which orders the two events with respect to each other. The ordering constraint implied by this relation might be useful if, for example, the following statement was part of the knowledge base.

(42) The Chinese takeover of Hong Kong took place on July 1, 1997.

Another way that partial information about the temporal extent of events is expressed in English is the tense/aspect system. This grammatical marking system expresses the temporal extent of an event expressed by a finite clause with respect to the time of utterance (in the case of texts, the document creation time), and to a reference time (for details, see (Reichenbach, 1947)). This can be seen in examples such as the following:

(43) a. Kidnappers *kept* their promise to kill a store owner they took hostage ...

b. The killers had *demanded* a ransom when they *contacted* police on Tuesday.

In (43a), the use of the past tense indicates that the event *kept* is located before the document creation time. In (43b), the use of perfective aspect indicates that the event *demanded* was complete before the time of the event *contacted*.

We need to consider what relations our language needs to express in order to retrieve answers to questions. (Allen, 1984) laid out the space of possibilities for relating intervals to one another based on the possible ordering of the endpoints of intervals, as shown in Figure 1.

English does not succintly express all of the possible relations between intervals, so it might not be necessary for a temporal annotation language to use all of them. *Overlaps*, for example, is difficult to find instantiated in natural language text. The most important requirement of the set of relations used by the language under consideration is that it allow inferences to be drawn from pairs of relations. This is because in natural language texts, it is uncommon for each event to be associated with a time. As we have seen, much more significant is the placement of events with respect to each other and with respect to certain key times. That is, information about the placement of a given event on the timeline is almost always partial and distributed across several clauses. Thus, a QA system needs to have reference to a system of relations which allows it to easily combine the different

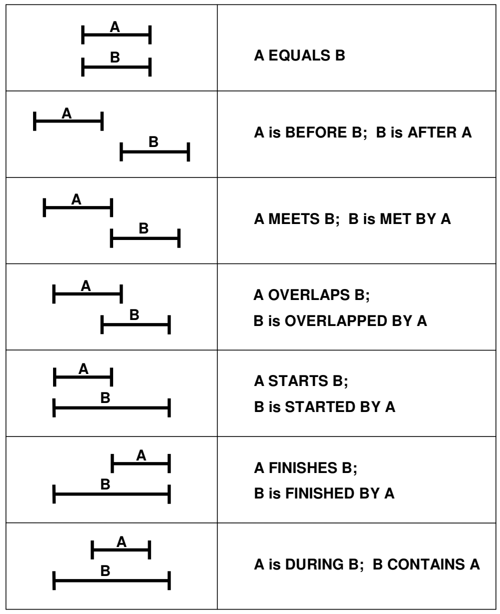

Figure 1. The interval relations as defined by Allen (1984)

kinds of temporal information expressed in a set of English statements and make judgements about the temporal location of events. Because reasoning over the Allen relations is well-understood, they provide a good basis for the set of relations needed by a QA system. Further, the more temporally-oriented questions seen in Section 2 are phrased directly in terms of temporal relations.

3.1.4. *Subordination relations*

Examination of the following examples makes clear that there is, in fact, another kind of event-event relation that is important to be able to represent in a QA knowledge base.

(44) a. Five other U.N. inspection teams *visited* a total of nine other sites, the agency reported.

b. . . . said he regretted the civilian *casualties* . . .

c. U.S. officials claim they already *see* signs Saddam Hussein is getting nervous.

d. German law requires parties to *publish* the name and address of anyone who donates . . . ...

The veracity of the event referred to by the italicized word in each example—whether the event can be treated as real—is affected by the fact that it is embedded under the underlined verb. The sentence does not simply represent the event as being part of the actual past or present, or projected future. Instead, it expresses the event in qualified terms. This is very similar to modality, discussed above. In (44a), for instance, the underlined event is qualified by being the argument of *report*. Its veracity depends on the reliability of the reporting agent. In (44b), the underlined event is presupposed to be true, because *regret* is a factive predicate in English (Kiparsky and Kiparsky, 1970). The relations expressed between subordinated events and the events that subordinate them are not temporal relations, per se, (though they may have temporal implications); nevertheless, it is crucial that they are represented in a QA knowledge base. In order to effectively answer a question about an event it is very important to know whether the writer has presupposed its veracity, deferred responsibility for its veracity to another party, or presupposed its falsity. The use of such relations and even their significance will vary from application to application, but a QA retrieval application will have to know that the relation exists. Thus, a language for modelling temporal information in texts should have some way to represent the different sorts of subordination relations that can be expressed. In Section 4, we present a complete set of relations for this purpose.

3.2. Extracting information for use by QA systems

As seen above, English (as well as any other natural language) has many mechanisms for expressing temporal properties of events. So far, we have primarily discussed the nature of those temporal properties themselves; we have considered what can be expressed from the perspective of how that information could be represented in such a way that a QA retrieval system could have access to it. Now we turn to considerations of how temporal information might be extracted from natural language texts. A temporal annotation language should also capture the kind of information that might be required by tools and algorithms for automatically annotating temporal information. Such tools and algorithms include machine learning techniques which might use human-annotated documents as training data, as well as more rule-based techniques, which might exploit linguistic regularities to derive temporal relations. We envision a multi-step, layered, information extraction process in which distinct modules may be responsible for extracting different pieces of information and incrementally marking up a knowledge base so information can be retrieved from it. Because the temporal relations representing the anchoring and ordering of events are the ultimate goal of an extraction process, such a process requires a practical way to represent all the building blocks which might be used to determine the temporal relations.

3.2.1. *Morphosyntax of events*

As discussed above, the tense/aspect system of English is an important method of locating events on the timeline. Thus, a language used to record information relevant to temporal information extraction systems would have to record the tense and aspect of finite verbs so it could infer a relation between the document creation time and the event based on the event's tense. While we would not expect a QA retrieval system to use information about the tense of a verb used to express a given event, tense/aspect information is absolutely necessary for any information extraction system which attempts to anchor events to times. It also seems likely that information about tense and aspect features may be an important component in attempts to extract temporal relations between events. Some positive results have been reported by (Grover et al., 1995), (Song and Cohen, 1991), and (Mani et al., 2003) using machine learning techniques for predicting temporal relations between events based on tense and aspect (among other morphosyntactic factors).

As mentioned above, many events are not expressed by finite verbs. Nonfinite verbs are quite common ways to refer to events, and nouns and adjectives are not insignificant. (Lapata and Lascarides, 2004) found that the grammatical category of nonfinite verbs that express events is significant in predicting temporal relations between events which occur in the same sentence. We expect that the part-of-speech of events not expressed with verbs may be significant in drawing inferences about the temporal anchoring of events. For example, adjectives normally encode states, which, when they can be located temporally, are considered events for our purposes. States are fairly unique in that they are generally persistent. This property of persistence may well be usable to make temporal inferences; an introduced state can be assumed to continue unless its termination is explicitly mentioned. So the part-of-speech of an event-expressing term may give clues as to the type of event, which might in turn have implications about the temporal properties of the event. Next, we turn to the importance of such a typology of events for temporal information extraction.

3.2.2. *Typology of events*

There are several different ways to categorize events. The just-mentioned distinction between states and events is based on temporal properties of events, and is part of the (Vendler, 1957) classification, which focusses on the internal temporal structure of events. It is certainly plausible that a Vendlerian classification of events might provide useful input to algorithms which attempt to anchor and order events in text, and a temporal annotation language could certainly adopt this classification. However, other aspects of Vendler's scheme involve quite a bit more sublety than the event/state distinction, and the implications of the distinctions for inferring temporal relations are not at all straightforward. As we discuss in Section 4, the event/state distinction is the only aspect of Vendler's classification that we adopt. We are, however, concerned with having the ability to automatically infer or extract the sort of embedding relations mentioned above in 3.1.4. It seems that the sort of embedding relation involved is easily predictable from semantic features of the verb. Consider the following examples of events which might be intuitively classed as 'reporting' events of some sort.

(45) a. In the air, U.S. Air Force fliers <u>say</u> they have *engaged* in ...

b. In Kuwait, the Iraqis have *rimmed* the capital city with an air-defense system, <u>according</u> to a U.S. official.

c. A senior law enforcement source <u>tells</u> CNN, the evidence is mostly *circumstantial*.

d. At least 51 people were <u>reported</u> *killed* in clashes between Serb police and ethnic Albanians.

e. The spokesman <u>added</u> that the deal has not been *signed* yet.

It is easy to see that these underlined reporting events bear roughly the same sort of relation to their embedded events as described in 3.1.4 They seem to express that the agent of the reporting event is responsible for establishing the veracity of the embedded event. It would seem that a classification which recognized these verbs as belonging to the same class would allow us to easily infer the presence of this particular type of subordination relation. Thus, it would be very useful for a temporal information extraction system to have access to a classification of events which would allow it to predict the sort of subordination relations they introduce, particularly in case there is some level of ambiguity in the way such relations are introduced; if a given verb can have senses which introduce subordination relations and senses which do not, a classification algorithm will be needed in order to automatically infer the presence of these relations. For example, the verb *add*, which is used in (45e) as a reporting verb, obviously has senses in which it has no reporting meaning and thus does not introduce a subordination relation. This suggests that extracting subordination relations will be a multi-step process which involves the classification of events by some sort of disambiguation. In section 4, we propose a set of event types for this purpose.

One particular class of events which subordinate other events deserves special mention here, because it introduces subordination relations of a unique sort.

3.2.3. *Aspectual verbs and relations*
Consider the examples below.

(46) a. The tank began *leaking* oil on Friday morning.

b. The phony *war* has finished and the real referendum campaign has clearly begun.

c. An intense *manhunt* continues for Rudolph in the wilderness of western North Carolina.

In each, the underlined verb, rather than expressing an event as such, expresses an important temporal property of the event referred to by the embedded verb. We might say that it 'chooses' an event or a time and expresses a temporal relation between that event or time and the underlined event. For example, in (46a), the event of leaking is said to begin on Friday. If we compare the sentence in (47), we see that different relations between the time *Friday morning* and the event *leaking* are expressed in these cases. (46a) gives more specific information about the temporal extent of the *leaking* event.

(47) The tank was *leaking* oil on Friday morning.

In the former, a *begins* relation is expressed between *leaking* and *Friday morning*. Without the verb *began*, only an *includes* relation is expressed between the two, meaning the temporal extent of *leaking* includes the time referred to by *Friday morning*. Thus, the fact that *leaking* is subordinated to *began* in this example is a very important fact that a temporal annotation language must record if it is going to be useful for inferring anchorings and orderings of events. The behavior of verbs like *begin*, *stop*, *continue*, etc., closely parallels the grammatical category of aspect, and thus, the underlined verbs are often referred to as 'aspectual' verbs. In Section 4, we introduce a classification of events which includes aspectual events. While the temporal extent of aspectual events are not in themselves of interest, they seem to form a class in that they introduce a special sort of subordination relation. Paralleling the event classification, we refer to the sort of subordination relation they introduce as an 'aspectual' relation. Note that aspectual verbs can be subcategorized in terms of the sort of aspectual relation they introduce. For example, verbs like *begin*, *start*, and *commence* all express an aspectual relation which allows one to infer a *begins* relation between the embedded verb and some other time or event. A temporal annotation language should both recognize the category of aspectual verbs, and provide a way to characterize the subordination relations they introduce.

3.2.4. *Signals of temporal relations*

Above, we saw examples of temporal relations being expressed by temporal prepositions and conjunctions like *before*, *after*, *while*, *on*, etc. In order to automatically extract temporal relations, it is important to be able to first identify such signals. Because these expressions have multiple uses (for example, *on* can be used as a locative preposition), it becomes necessary to identify when they are signalling a temporal relation and when they are not. Again, we envision a multi-step information extraction process, in which temporal signals are likely to be identified early so later algorithms can exploit them. Because we also expect that human-annotated corpora will be used for machine learning, it will be necessary for the markup language to have some way to associate temporal relations with the signals that express them.

3.2.5. *The functional content of time expressions*

As seen above, time expressions in natural language do not usually fully specify a time. Instead, they often function indexically, picking out a time via reference to some anchoring time in the context, as seen below:

(48) a. The White House press secretary reports that the president will leave for Istanbul *tomorrow*.

b. The prime minister's last visit was *in October*.

c. He didn't make it to Istanbul until *the following Saturday*.

Tomorrow, for example, does not refer to any particular time until its indexical anchor (usually the document creation time) is recovered. It refers to a day one day after that anchor. As a time expression on its own, a phrase like *October* has similar behavior. The October it picks out is picked out with respect to the document creation time. Expressions like *the following Saturday* parallel this behavior, except that their anchors are times other than the document creation time. It is possible to see such expressions as functional in the sense that they return determinate values based on their anchoring. That is, their meaning returns a value when given an anchor time. While we assume that a QA retrieval system will have more use for the fully-specified, normalized value of a time expression, the possibility of representing the functional content which is the meaning of these expressions would be extremely useful in an incremental process for extracting temporal information; it allows the process of recovering the functional content of these expressions to be separated from the process of normalizing and fully specifying their value. In Section 4, we present a proposal for representing the functional content of time expressions.

4. TimeML

The questions presented in the question corpus revealed that some understanding of time was necessary in order to both model and answer the questions. Moreover, many of the questions and the data that could answer them involved temporal relationships in an implicit way. That is, while some questions such as *When was John F. Kennedy president?* require the use of time directly, others are far less explicit. For example, the question, *Who was president in 1958?* is not so directly about time (i.e. it is not a *when* question), but it surely requires a temporal understanding to answer it.

In Section 3, the features of a system capable of working in a QA system were discussed. Such a system must be able to represent temporal expressions, events, and relationships. TimeML is a modeling language that has been designed with these features in mind. In this section, we discuss how this is accomplished and point out some of the expressive power of TimeML.

The tags employed in TimeML are all intended to assist in the understanding of time so that questions and corpora can be modeled, leading to eventual question answering. To that end, TimeML used four different tag types. The `TIMEX3` tag is used to capture all temporal expressions. The `EVENT` tag captures all temporal events. Functional words such as *at* and *from* are annotated with the `SIGNAL` tag. Finally, all relationships between the other tags are represented with the `LINK` tags: `TLINK`, `SLINK`, and `ALINK`. For a complete description of TimeML, the reader can refer to www.timeml.org.

4.1. Representing Temporal Expressions

At the core of any scheme designed to provide temporal understanding is a method for representing specific temporal expressions such as *1961* or *today*. TimeML models this type of expression with the TIMEX3 tag. There are four types of temporal expressions captured in TIMEX3: TIME, DATE, DURATION, and SET, each corresponding with the types described in 3.1.2.

An expression that receives the **TIME** type is one that refers to a time of the day, even if in a very indefinite way. The easiest way to distinguish a TIME from a DATE is to look at the granularity of the expression. If the granularity of the expression is smaller than a day, then the expression is a TIME. For example, the following expressions fit into this category:

Mr. Smith left **ten minutes to three**
at **five to eight**
at **twenty after twelve**
at **half past noon**
at **eleven in the morning**
at **9 a.m. Friday, October 1, 1999**
the morning of January 31
late last night

Notice that most of these examples are not fully specified temporal expressions. That is, they appear to be within a context that provides their complete specification, including the date on which they take place. With the exception of the expression *9 a.m. Friday, October 1, 1999*, each of these expressions requires more information to fully represent what they entail. This is a recurring phenomenon with temporal expressions that TimeML addresses with temporal functions. This technique will be discussed shortly.

The **DATE** type can be thought of as any expression that refers to a calendar time. Again, there may be some confusion as to when an expression is a TIME and when it is a DATE. The granularity test continues to help with this as DATEs are generally of a day or larger temporal unit. As with TIMEs, DATEs are often underspecified. Here are a few examples:

Mr. Smith left **Friday, October 1, 1999**
the second of December
yesterday
in **October of 1963**
in **the summer of 1964**
on **Tuesday 18th**
in **November 1943**
this year's summer
last week

An expression is a **DURATION** if it explicitly describes some extent of time. Examples of this are:

Mr. Smith stayed **2 months** *in Boston*
48 hours
three weeks
all last night
20 days *in July*
3 hours *last Monday.*

Finally, the **SET** type is used for expressions that describe a set of regularly reoccurring times. These are expressions such as:

John swims **twice a week**.
every 2 days.

The type of a temporal expression is represented in the tag along with a specific value for the time expression. A temporal expression's value is annotated with an extension of the ISO 8601 standard. For example, a fully specified temporal expression such as the one in (49a) has a `value` of "2004-11-22". A TimeML annotation produces XML as in example (49b).

(49) a. November 22, 2004

b.
```
<TIMEX3 tid="t1" type="DATE" value="2004-11-22">
November 22, 2004
</TIMEX3>
```

The `value` of temporal expressions that are not fully specified are not as obvious as those whose extent contains all of the necessary information. For these kinds of expression, the value must be normalized. But, before this is discussed, it is useful to examine one more aspect of the simple example in (49). The `tid` attribute is an automatically assigned ID number that allows the expression to be mentioned elsewhere in the annotation. For instance, "t1" above might participate in a temporal link with some event. The method for doing this is found in 4.4.1, but it is enough to say, for now, that all objects in TimeML receive an ID number similar to the `tid` given in the `TIMEX3` tag.

When a temporal expression is not fully specified, placeholders can be used in the `value` attribute. For example, an expression such as *January 12* provides no year information. It can be given a `value` of *XXXX-1-12*. In the case of times and dates, these placeholders are generally removed in favor of a more complete annotation provided by temporal functions. Durations and sets are rarely, if ever, underspecified, but they do receive some special attention in both the `value` attribute and the `TIMEX3` tag as a whole. In the following subsection, temporal functions for times and dates will be described, but, first, we will briefly turn to these special aspects.

The first attribute value of note for durations is contained in `value`. Durations are required to a have a particular format in this attribute because they represent a period of time. A sample annotation for a simple duration is given in (50).

(50)
```
<TIMEX3 tid="t1" type="DURATION" value="P3D">
three days
</TIMEX3>
```

Durations are also elligible to use two additional `TIMEX3` attributes: `beginPoint` and `endPoint`. These are used to capture what were called in Section 3 *anchored durations*. For example, the expression *a week from Monday* has a begin point, namely, the `tid` for *Monday*. With this information, the actual date that the full phrase refers to can be calculated. TimeML allows for an additional `TIMEX3` to be created to annotate the missing point. This is a useful and neccesary part of TimeML. The following example reveals why.

(51) John will leave a week from Monday.

Althought we have not yet introduced the TimeML methods for capturing events and temporal relationships, it should be clear that *leave* is linked in some way to the expression *a week from Monday*. Yet, it is not directly related to either *a week* or *Monday*. Using the method described above, a `tid` can be created that can participate in a link such that *leave* is truly anchored to the correct time.

In the case of the SET type, the `value` attribute must work together with at least one of two additional `TIMEX3` attributes: `quant` and `freq`. The former represents any quantifier that is used in the expression. For instance, *every Tuesday* would receive a `quant` of *EVERY* and a `value` of "XXXX-WXX-2", the ISO 8601 representation of *Tuesday*. The frequency of the expression is represented in the `freq` attribute as in *3 days each week*. The annotation of this expression is given in (52).

(52)
```
<TIMEX3 tid="t1" type="SET" value="P1W" quant="EACH"
freq="3D">
3 days each week
</TIMEX3>
```

4.1.1. *Functional Content of Temporal Expressions*
TimeML strives to capture all temporal expressions with the `TIMEX3` tag, but, as is apparent in the above examples, many of these expressions seem to be missing information critical to their full specification. In fact, analysis of the corpus reveals that there are generally very few fully specified temporal expressions. The reader uses these to fully appreciate the rest of the temporal expressions. Temporal functions are TimeML's way of doing the same thing. When a `TIMEX3` is underspecified, it is anchored to a fully specified temporal expression. This is often the expression that includes the `functionInDocument` attribute in its `TIMEX3`. For example, a news report often includes a specific document creation time. If the article refers to *today*, that expression is anchored to the document creation time to complete its specification. In the same manner, an expression such as *July 9* is underspecified until the appropriate year is supplied. Since that information can be extracted from the document creation time, it is anchored to that `TIMEX3` and the correct year is added to the value of the *July 9* `TIMEX3`.

When an expression requires an anchoring to be completely specified, an attribute called `temporalFunction` receives a "true" value. When an annotation is done manually, this attribute is just an indication that the value of the `TIMEX3` was calculated by way of a temporal anchor, which the annotator must also supply. An automatic annotation will use functions to do the same thing. In the next sections, descriptions of these functions are provided along with examples of the functions in action. Notice that the underspecified `TIMEX3s` still have three core attributes: `tid`, `type`, and `value`. When a temporal function is also used, three more attributes are added:

- `temporalFunction` – a boolean attribute that indicates that a function is necessary
- `anchorTimeID` – the `tid` of another `TIMEX3` that provides information to the temporal function
- `valueFromFunction` – the `tfid`, or temporal function ID, of the function that completes the `TIMEX3` specification

The reader may wonder why both the `value` and `valueFromFunction` attributes are used since expressions that require functions, by definition, do not contain enough information to provide a value. However, it is not always the case that the expression doesn't contain any specific temporal information at all. In cases such as *today*, the extent of the tag cannot lend any information to the `value` attribute and, truly, the temporal function must do all the work. Still, cases such as *Wednesday* do contain specific information that should be captured by the `TIMEX3` tag. In the former case, the `value` must be something like "XXXX-XX-XX", where the X-placeholder is used to show that the format of this value should be that of a DATE, but that no other information has been provided. In the latter case, though, it is useful to capture that the expression makes use of specific temporal information by giving a `value` of "XXXX-WXX-3".

4.1.2. *Specification of Selected Temporal Functions*

1. `<IndefiniteFuture tfid= argumentID=  />`

   Usage: Indicate a future reference

   `argumentID`: ID of last anchor in the chain of functions

   Example: *in the <u>future</u>*

```
<TIMEX3 tid="t1" type="DATE" value="FUTURE_REF"
   temporalFunction="true" valueFromFunction="tf1"
   anchorTimeID="t0">
future
</TIMEX3>
<IndefiniteFuture tfid="tf1" argumentID="t0"/>
```

2. `<IndefinitePast tfid= argumentID= />`

   Usage: Indicate a past reference

   `argumentID`: ID of last anchor in the chain of functions

   Example: *in the <u>past</u>*

```
<TIMEX3 tid="t1" type="DATE" value="PAST_REF"
   temporalFunction="true" valueFromFunction="tf1"
   anchorTimeID="t0">
past
</TIMEX3>
<IndefinitePast tfid="tf1" argumentID="t0"/>
```

3. `<Identity tfid= argumentID= />`

   Usage: Indicate a present reference

   `argumentID`: ID of last anchor in the chain of functions (i.e. whatever time "now" refers to)

   Example: *<u>now</u>*

```
<TIMEX3 tid="t1" type="DATE" value="PRESENT_REF"
   temporalFunction="true" valueFromFunction="tf1"
   anchorTimeID="t0">
now
</TIMEX3>
<Identity tfid="tf1" argumentID="t0"/>
```

4. `<CoerceTo tfid= argumentID= scale= />`

   Usage: Returns the enclosing time period of the specified type given in `scale`

   `argumentID`: ID of last anchor in the chain of functions; generally, the DCT ID for simple temporal expressions

   `scale`: Name of a type of time period (granularity); "hour, minute, day, year", etc.

   Example: *this week*

```
<TIMEX3 tid="t1" type="DURATION" value="P1W"
   temporalFunction="true" valueFromFunction="tf1"
   anchorTimeID="t0">
this week
</TIMEX3>
<CoerceTo tfid="tf1" argumentID="t0" scale="WEEK"/>
```

5. `<Predecessor/Successor tfid= argumentID= count= signalID= />`

   Usage: Given a time period of a standard granularity, returns a new time period of the same type that precedes or succeeds the original by the number given in `count`

   `argumentID`: ID of last anchor in the chain of functions

   `count`: Numeric attribute that specifies how much to move on the timeline

   `signalID`: ID of the signal that prompted the use of the function

   Example: *4 weeks ago*

```
<TIMEX3 tid="t1" type="DURATION" value="P4W"
   temporalFunctiion="true" valueFromFunction="tf1"
   anchorTimeID="t0">
4 weeks
</TIMEX3>
<SIGNAL sid="s1">
```

```
ago
</SIGNAL>
<CoerceTo tfid="tf2" argument="tf1" scale="WEEK"/>
<Predecessor tfid="tf1" argument="tf2" count="4" signalID="s1"/>
```

6. `<Adjust tfid= argumentID= signalID= direction= quantity= value= />`

   Usage: Indicates a modification of the argument time or time period; an approximation function

   `argumentID`: ID of last anchor in the chain of functions

   `signalID`: ID of SIGNAL that prompted the use of the function

   `direction`: `later|earlier` (for times), `larger|smaller` (for time periods), `unspecified` (for adjustments in either direction)

   `quantity`: a numeral, `unspecified`, or `small` that indicates the amount of adjustment

   `value`: the value of the argument time or time period – this information is captured in the `TIMEX3` tag, so this attribute should likely be dropped

   Example: <u>*for just over two years*</u>

```
<TIMEX3 tid="t1" type="DURATION" value="P2Y"
   temporalFunction="true" valueFromFunction="tf1">
for just over two years
</TIMEX3>
<Adjust tfid="tf1" argumentID="t1" direction="larger" quantity="smaller"/>
```

4.2. Representing Events

The goal of TimeML is to provide a language for the representation of temporal relations. Temporal expressions, captured with `TIMEX3`, are the first ingredient in many of these relationships. Events are the next ingredient and are primarily represented with the `EVENT` tag, followed by the `MAKEINSTANCE` tag.

4.2.1. *The EVENT Tag*

Much like the `TIMEX3` tag, TimeML captures several different types of event. The type of event is stored in the `class` attribute. A TimeML event will fit into one of these categories:

1. REPORTING:

   When a person or organization declares something, narrates an event, or informs about an event, the event that describes that action is of the REPORTING class. These are generally verbs such as: *say, report, tell, explain, state.*

2. PERCEPTION:

   This class includes events that involve the physical perception of another event. Such events are typically expressed by verbs like: *see, watch, glimpse, behold, view, hear, listen, overhear.*

3. ASPECTUAL

   In languages such as English and French, there is a grammatical device of aspectual predication, which focuses on different facets of event history:

   a) Initiation: *begin, start*
   b) Reinitiation: *restart, reinitiate, reignite*
   c) Termination: *stop, cancel*
   d) Culmination: *finish, complete.*
   e) Continuation: *continue*

   Events that are of this class also participate in a particular kind of TimeML link called an `ALINK` (for "Aspectual Link") so that the relationship between the ASPECTUAL event and the one it predicates over can be shown.

4. I_ACTION:

   An I_ACTION is an Intensional Action. An I_ACTION introduces an event argument, which must be in the text explicitly. The event argument describes an action or situation from which we can infer something given its relation with the I_ACTION. For instance, the events introduced as arguments of some I_ACTIONS may not necessarily have occurred when the I_ACTION takes place. Explicit performative predicates are also included here. Note that the I_ACTION class does not cover states as they have their own associated classes.

   For the most part, events that are tagged as I_ACTIONs are in a closed class. The following list provides a sampling of this class:

   a) attempt, try, scramble

b) investigate, investigation, look at, delve
c) delay, postpone, defer, hinder, set back
d) avoid, prevent, cancel
e) ask, order, persuade, request, beg, command, urge, authorize
f) promise, offer, assure, propose, agree, decide
g) swear, vow
h) name, nominate, appoint, declare, proclaim
i) claim, allege, suggest

5. I_STATE

   I_STATE events are similar to the previous class. This class includes states that refer to alternative or possible worlds (delimited by square brackets in the examples below), which can be introduced by subordinated clauses (a), nominalizations (b), or untensed VPs (c):

   a) *Russia now* **feels** [*the US must hold off at least until UN secretary general Kofi Annan visits Baghdad*].
   b) *"There is no reason why we would not be* **prepared** *for [an attack"].*
   c) *The agencies fear they will be* **unable** *[to crack those codes to eavesdrop on spies and crooks].*

   Here again is a list of events that fall into this category:

   a) believe, think, suspect, imagine, doubt, feel, be conceivable, be sure
   b) want, love, like, desire, crave, lust
   c) hope, expect, aspire, plan
   d) fear, hate, dread, worry, be afraid
   e) need, require, demand
   f) be ready, be eager, be prepared
   g) be able, be unable

6. STATE:

   STATEs describe circumstances in which something obtains or holds true. However, only certain events in this category are annotated in TimeML:

a) States that are identifiably changed over the course of the document being marked up. Remember that TimeML's chief concern is to annotate temporal events. If a STATE is deemed persistent throughout the event line of the document, it is factored out and not annotated. Conversely, if a property is known to change during the course of events represented or reported in the article, that property is marked as a STATE.

b) States that are directly related to a temporal expression.
If a STATE directly participates in a temporal relationship, it must be annotated to do so. Again, this is an example of limiting TimeML STATEs to ones that involve time.

c) States that are introduced by: an I_ACTION, an I_STATE, or a REPORTING event.

d) Predicative states the validity of which is dependent on the document creation time.

7. OCCURRENCE:

This class includes all the many other kinds of events describing something that happens or occurs in the world. Essentially, this is a catch-all category for events that participate in the temporal annotation, but do not fit into any of the above categories.

The annotation of an `EVENT` is quite simple as it only includes the `class` attribute and a tag that identifies it. The following tag holds much more information about the event, or rather an instance of that event. As such, examples of annotated `EVENTs` are provided below.

4.2.2. *The MAKEINSTANCE Tag*

Once an event is tagged in TimeML, an instance of that event is created with the `MAKEINSTANCE` tag. It is this event instance that participates in temporal relationships. `MAKINSTANCE` is the first example of a non-consuming TimeML tag. That is, both the `TIMEX3` and `EVENT` tags are inserted directly into a document so they surround the text they capture. Again, the data calls for instances of an event to be annotated out of line because these instances do not always capture text directly from the document. This tag was developed to capture multiple instances of an event. The following simple sentence reveals why `MAKEINSTANCE` is necessary in this case:

"John teaches on Monday and Wednesday."

One might believe the `EVENT` and `TIMEX3` tags along with the soon to be discussed temporal relationship tags could successfully capture the information this sentence contains. However, without multiple instances of the *teaches* event, such a relationship would suggest that the *same* event occurs on both Monday and Wednesday. The MAKEINSTANCE tag allows a more accurate representation of this sentence such that the occurrences of *teaches* on Monday and Wednesday are unique. In the above sentence, the *teaches* event is annotated first with an EVENT tag and then with two MAKEINSTANCE tags:

```
John
<EVENT eid="e1" class="OCCURRENCE">
teaches
</EVENT>
on Monday and Wednesday.
<MAKEINSTANCE eiid="ei1" eventID="e1" tense="PRESENT"
aspect="NONE"/>
<MAKEINSTANCE eiid="ei2" eventID="e1" tense="PRESENT"
aspect="NONE"/>
```

Along with this ability to create multiple instances of an event, the MAKEINSTANCE tag captures other information about the event instance. The values for these attributes are generally lexically motivated. The tense and aspect of the event are represented in the appropriate MAKEINSTANCE for that event along with the instance's modality and polarity. Again, this information must appear in a non-consuming tag because it can change for multiple instances of the event. A simple sentence can demonstrate this:

"John teaches on Monday but might not on Tuesday."

Here, one instance of *teaches* contains both a modal and negation operator while the other does not:

```
John
<EVENT eid="e2" class="OCCURRENCE">
teaches
</EVENT>
on Monday but might not on Tuesday
<MAKEINSTANCE eiid="ei1" eventID="e2" tense="PRESENT"
aspect="NONE"/>
<MAKEINSTANCE eiid="ei2" eventID="e2" tense="PRESENT"
aspect="NONE" modality="MIGHT" polarity="NEG"/>
```

4.3. Representing Signals

Functional words such as *before* and *during* are captured with the SIGNAL tag to make explicit the part they play in determining relationships between times and events. Once a text has been given an annotation for times, events, and signals, TimeML can begin to relate them to each other. In the next section, these relationships will be described. A complete annotation for a simple sentence, prior to any links being added, looks like this:

```
John
<EVENT eid="e2" class="OCCURRENCE">
teaches
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e2" tense="PRESENT"
aspect="NONE"/>
<SIGNAL sid="s1">
at
</SIGNAL>
<TIMEX3 tid="t1" type="TIME" value="2004-11-22T15:00"
temporalFunction="TRUE" anchorTimeID="t2">
3:00
</TIMEX3>
<SIGNAL sid="s2">
on
</SIGNAL>
<TIMEX3 tid="t2" type="DATE value="2004-11-22">
November 22, 2004
</TIMEX3>.
```

4.4. Representing Relationships

TIMEX3 and EVENT tags only begin to reveal the representational power of TimeML. In order to adequately represent text and queries for question answering, an annotation requires a method for capturing all sorts of temporal relationships as well as other relationships that have already been touched upon.

Many events are explicitly anchored to a specific time within a document. An article might include a sentence such as:

"John taught at 3:00 p.m."

In this case, the *taught* event can be stamped with the *3:00 p.m.* time so that this anchoring relationship is clear. Time stamping is an effective way to represent some temporal relationships, but it cannot capture relationships that involve the ordering of events and times, or any other relationships between two events. For example, the subordinating relationship an I_ACTION has with another event is key to the understanding of the text. That relationship may be a modal one that calls into question whether the latter event actually takes place, or it could negate that latter event altogether.

With instances of events available along with the annotated temporal expressions, TimeML can effectively do time stamping with a `LINK` tag, presented in the following section. TimeML is not limited to this kind of temporal relationship, though. The `LINK` tags capture both anchoring and ordering relationships as well as subordinating and aspectual ones between event instances.

There are three `LINK` tags in TimeML:

1. **TLINK**: Temporal Link, captures anchoring and ordering relationships

2. **SLINK**: Subordinating Link, captures subordinating relationships between event instances

3. **ALINK**: Aspectual Link, captures aspectual relationships between ASPECTUAL event (instances) and the event instance over which it predicates

As with the `MAKEINSTANCE` tag, these linking tags appear offline since they don't specifically capture any text. Each tag has particular attributes associated with it. The most crucial of these is the `relType` attribute, which has different possible values depending on the type of the link. Since the `relType` is the primary indicator for what relationship the participating temporal entities share, this attribute will be the focus of the following discussion of each tag.

4.4.1. *Temporal Links*

A `TLINK` or Temporal Link represents the temporal relationship holding between events, times, or between an event and a time. Note that `EVENTs` participate in a `TLINK` by means of their corresponding event instance IDs. In the present explanation, however, the words "events" and "event instances" are used interchangeably. This same observation applies also for `SLINKs` and `ALINKs`, below. As a rule, `EVENTs` never participate in a `LINK`. Only their associated event instances are eligible.

The following enumeration describes the possible values for the `relType` attribute in a `TLINK` tag:

1. **Simultaneous**

   Two event instances are judged simultaneous if they happen at the same time, or are temporally indistinguishable in context, i.e. occur close enough to the same time that further distinguishing their times makes no difference to the temporal interpretation of the text.

2. One **before** the other:

   As in the following example between the events *slayings* and *arrested*:

   > *The police looked into the* **slayings** *of 14 women. In six of the cases suspects have already been* **arrested***.*

3. One **after** the other:

   This is just the inverse of the preceding relation. So the two events of the previous example can alternatively be annotated as expressing an **after** relation, if the directionality is changed.

4. One **immediately before** the other:

   As in the following sentence between *crash* and *died*.

   > *All passengers* **died** *when the plane* **crashed** *into the mountain*

5. One **immediately after** than the other:

   This is the inverse of the preceding relation.

6. One **including** the other:

   As is the case between the temporal expression and the event in the following example:

   > *John* **arrived** *in Boston* **last Thursday***.*

7. One **being included** in the other:

   The inverse relation to the preceding one.

8. One holds **during** the other:

   Specifically applicable to states or events that persist throughout a duration, for example:

   > *James was* **CTO** *for* **two years***.*
   > *John* **taught** *for* **20 minutes** *on Monday.*

9. One being the **beginning** of the other:

   As holds between the first of the temporal expressions and the event in the following example:

   John was **in the gym** *between* **6:00 p.m.** *and 7:00 p.m.*

10. One being **begun by** the other:

    The inverse relation to the one just introduced.

11. One being the **ending** of the other:

    John was **in the gym** *between 6:00 p.m. and* **7:00 p.m.**.

12. One being **ended by** the other:

    The inverse relation to the one just introduced.

13. Event **identity**:

    Event identity is also annotated via the TLINK. The relationship is used when two events are deemed to be the same event within the document. E.g.:

    John **drove** *to Boston. During his* **drive** *he ate a donut.*

With this rich library of possible temporal relationships, the TLINK can both anchor an event instance to a particular time and order event instances with respect to one another. In addition, some of these relationships work specifically with events of the DURATION type. TLINK is arguably the most important tag in all of TimeML. It greatly increases the power of the annotation by providing the tools for temporal ordering, a feature lacking in traditional time stamping procedures. Whether a question itself requires ordering in its representation or the text that can answer that question necessitates it, the anchoring and ordering capabilities of the TLINK tag greatly increase the likelihood that question answering can be achieved.

To see these TLINKs in action, we complete the example from the end of the last section, adding two temporal links:

```
John
<EVENT eid="e2" class="OCCURRENCE">
teaches
</EVENT>
<MAKEINSTANCE eiid="ei1" eventID="e2" tense="PRESENT"
aspect="NONE"/>
<SIGNAL sid="s1">
```

```
at
</SIGNAL>
<TIMEX3 tid="t1" type="TIME" value="2004-11-22T15:00"
temporalFunction="TRUE" anchorTimeID="t2">
3:00
</TIMEX3>
<SIGNAL sid="s2">
on
</SIGNAL>
<TIMEX3 tid="t2" type="DATE value="2004-11-22">
November 22, 2004
</TIMEX3>.

<TLINK eventInstanceID="ei1" relatedToTime="t1" relType="IS_INCLUDED"
signalID="s1"/>
<TLINK timeID="t1" relatedToTime="t2" reltype="IS_INCLUDED"
signalID="s2"/>
```

4.4.2. *Subordinating Links*

An SLINK or Subordination Link is used for contexts introducing relations between two events. SLINKs are of one of the following sorts:

- **Modal:**

  This relation is brought up by events introducing a reference to a possible world – mainly I_ACTIONs and I_STATEs:

  John **promised** *Mary to buy some beer.*
  Mary **wanted** *John to buy some wine.*

- **Factive:**

  Certain verbs introduce an entailment (or presupposition) of their argument's veracity. They include **forget** (with a tensed complement), **regret**, or **manage**:

  John **forgot** *that he was in Boston last year.*
  Mary **regrets** *that she didn't marry John.*
  John **managed** *to leave the party*

- **Counter-factive:**

  Contrary to the previous relation, in this case the event introduces a presupposition about the non-veracity of its argument: **forget (to), unable to** (in past tense), **prevent, cancel, avoid, decline**, etc.

John **forgot** *to buy some wine.*
Mary was **unable** *to marry John.*
John **prevented** *the divorce.*

- **Evidential:**

  Evidential relations are typically introduced by REPORTING or PERCEPTION events:

  John **said** *he bought some wine.*
  Mary **saw** *John carrying only beer.*

- **Negative evidential:**

  Introduced by REPORTING and PERCEPTION events conveying negative polarity:

  John **denied** *he bought only beer.*

- **Conditional:**

  Introduced by the presence of an 'if' construction:

  If John buys only beer, Mary will get the wine.

SLINKs can be either lexically or structurally-based. Those that are lexically-based are SLINKs introduced by I_ACTION, I_STATE, PERCEPTION, and REPORTING events. These events generally take a clausal complement or a noun phrase headed by an event-denoting nominal. The SLINK is established between those events and the one denoted by the complement. An SLINK is always introduced when an event is tagged as being in one of these classes.

Structurally-based SLINKs are motivated by purpose clauses and conditional constructions. In the first case, an SLINK relates the event in the main clause (bold face) and the one in the purpose clause modifying it (underlined):

> *The environmental commission must* **adopt** *regulations to ensure people are not exposed to radioactive waste.*

In this example, *adopt* puts *ensure* in a modal context, motivating an SLINK with a MODAL reltype.

In a conditional construction, an SLINK relates the event in the antecedent clause and the one in the consequent clause:

> *On Dec. 2 Marcos promised to* **return** *to the negotiating table if the conflict zone was demilitarized.*

4.4.3. *Aspectual Links*

An ALINK or Aspectual Link represents the relationship between an aspectual event and its argument event. Examples of the aspectual relations to be encoded are:

1. **Initiation**:

   John **started** *to read*

2. **Culmination**:

   John **finished** *assembling the table.*

3. **Termination**:

   John **stopped** *talking.*

4. **Continuation**:

   John **kept** *talking.*

An ALINK is required whenever an event is classified as ASPECTUAL. They are, as such, completely motivated by the text.

When a document is fully annotated with each of these links, an accurate picture of the text takes shape with respect to time.

5. Conclusion

In this paper, we have presented a descriptive framework with which to examine the temporal aspects of natural language queries. We then demonstrated generally how tense and temporal information is encoded in language, motivating a particular specification description. The rest of the paper reported on work done towards establishing a broad and open standard metadata markup language for natural language texts, examining events, temporal expressions, and their orderings. What is novel in this language, TimeML, we believe, is the integration of three efforts in the semantic annotation of text: TimeML systematically anchors event predicates to a broad range of temporally denotating expressions; it provides a language for ordering event expressions in text relative to one another, both intrasententially and in discourse; and it provides a semantics for underspecified temporal expressions, thereby allowing for a delayed interpretation.

Significant efforts have been launched to annotate the temporal information in large textual corpora, according to the specification of TimeML described above. The result is a gold standard corpus, known as TIMEBANK, which has been released for general use. (Pustejovsky et al., 2003) We have also worked towards integrating TimeML with the DAML-TIme language (Hobbs, 2002, Hobbs and Pustejovsky, 2003), for providing an explicit interpretation of the markup described in this paper. It is hoped that this effort will provide a platform on which to build a multi-lingual, multi-domain standard for the representation of events and temporal expressions. We are currently building on this work to develop temporal awareness algorithms in the context of question answering systems within the AQUAINT program.

Acknowledgements

The authors would like to thank the other members of the TERQAS and TANGO Working Groups on TimeML for their contribution to the specification language presented here. In particular, we would like to thank Jerry Hobbs, Inderjeet Mani, Rob Gaizauskas, and Graham Katz. This work was performed in support of the Northeast Regional Reseach Center (NRRC) which is sponsored by the Advanced Research and Development Activity in Information Technology (ARDA), a U.S. Government entity which sponsors and promotes research of import to the Intelligence Community which includes but is not limited to the CIA, DIA, NSA, NIMA, and NRO. It was also funded in part by the Defense Advanced Research Projects Agency as part of the DAML program under Air Force Research Laboratory contract F30602-00-C-0168.

References

Abney, S., M. Collins, and A. Singhal: 2000, 'Answer Extration'. In: *Proceedings of the Applied Natural Language Processing Conference (ANLP)*. Seattle, WA, pp. 296–301.

Allen, J.: 1984, 'Towards a General Theory of Action and Time'. *Artificial Intelligence* **23**, 123–154.

Filatova, E. and E. Hovy: 2001, 'Assigning Time-Stamps to Event-Clauses'. In: *Proceedings of the 2001 ACL Workshop on Temporal and Spatial Information Processing.*

Grover, C., J. Hitzeman, and M. Moens: 1995, 'Algorithms for Analysing the Temporal Structure of Discourse'. In: *Proceedings of the Sixth International Conference of the European Chapter of ACL.*

Hovy, E., U. Hermjakob, and D. Ravichandran: 2002, 'A Question/Answer Typology with Surface Text Patterns'. In: *Proceedings of the Second International Conference on Human Language Technology Research (HLT 2002).* San Diego, CA, pp. 247–251.

Kiparsky, P. and C. Kiparsky: 1970, 'Fact'. In: *Progress in Linguistics.* The Hague: Mouton, pp. 143–173.

Lapata, M. and A. Lascarides: 2004, 'Inferring Sentence-Internal Temporal Relations'. In: *Proceedings of HLT-NAACL 2004.*

Mani, I., B. Schiffman, and J. Zhang: 2003, 'Inferring Temporal Orderings of Events in News'. In: *Proceedings of HLT-NAACL 2003.*

Mani, I. and G. Wilson: 2000, 'Robust Temporal Processing of News'. In: *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL2000).* New Brunswick, New Jersey, pp. 69–76.

Pustejovsky, J., L. Belanger, J. Castaño, R. Gaizauskas, P. Hanks, B. Ingria, G. Katz, D. Radev, A. Rumshishky, A. Sanfilippo, R. Saurí, A. Setzer, B. Sundheim, and M. Verhagen: 2002, 'TERQAS Final Report'. Technical report, The MITRE Corporation, Bedford, Massachusetts.

Pustejovsky, J., P. Hanks, R. Saurí, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, and M. Lazo: 2003, 'The TIMEBANK Corpus'. In: *Proceedings of Corpus Linguistics.* pp. 647–656.

Reichenbach, H.: 1947, *Elements of Symbolic Logic.* New York:Macmillan.

Schilder, F. and C. Habel: 2001, 'From Temporal Expressions To Temporal Information: Semantic Tagging Of News Messages'. In: *ACL-EACL-2001.* Toulose, France, pp. 65–72.

Setzer, A.: 2001, 'Temporal Information in Newswire Articles: an Annotation Scheme and Corpus Study'. Ph.D. thesis, University of Sheffield, UK.

Song, F. and R. Cohen: 1991, 'Tense interpretation in the context of narrative'. In: *Proceedings of AAAI '91.* pp. 131–136.

Vendler, Z.: 1957, 'Verbs and Times'. *Philosophical Review* **66**, 143–160.