Understanding Temporal Expressions in Emails

Benjamin Han, Donna Gates and Lori Levin
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh PA 15213
{benhdj|dmg|lsl}@cs.cmu.edu

Abstract

Recent years have seen increasing research on extracting and using temporal information in natural language applications. However most of the works found in the literature have focused on identifying and understanding temporal expressions in *newswire texts*. In this paper we report our work on *anchoring* temporal expressions in a novel genre, *emails*. The highly under-specified nature of these expressions fits well with our constraint-based representation of time, *Time Calculus for Natural Language* (TCNL). We have developed and evaluated a Temporal Expression Anchoror (TEA), and the result shows that it performs significantly better than the baseline, and compares favorably with some of the closely related work.

1 Introduction

With increasing demand from ever more sophisticated NLP applications, interest in extracting and understanding temporal information from texts has seen much growth in recent years. Several works have addressed the problems of representing temporal information in natural language (Setzer, 2001; Hobbs and Pan, 2004; Saurí et al., 2006), extracting and/or anchoring (normalizing) temporal and event related expressions (Wiebe et al., 1998; Mani and Wilson, 2000; Schilder and Habel, 2001; Vazov, 2001; Filatova and Hovy, 2001), and discovering the ordering of events (Mani et al., 2003). Most of these works have focused on capturing temporal information contained in *newswire texts*, and whenever both *recognition* and *normalization* tasks of temporal expressions were attempted, the latter almost always fell far behind from the former in terms of performance.

In this paper we will focus on a different combination of the problems: anchoring temporal expressions in *scheduling-related emails*. In our project work of building personal agents capable of scheduling meetings among different users[1], understanding temporal expressions is a crucial step. We have therefore developed and evaluated our system Temporal Expression Anchorer (TEA) that is capable of normalizing such expressions in texts. As input TEA takes English text with temporal expressions already identified, and transduces the expressions into their representations using *Time Calculus for Natural Language* (TCNL) (Han and Kohlhase, 2003). These representations, or TCNL formulae, are then evaluated by incorporating the contextual information to give the final normalized output. TCNL has the following characteristics: (1) a human calendar (e.g., the Gregorian calendar) is explicitly modeled as a constraint system to deal with the highly under-specified nature of many temporal expressions, and it allows easy extension to include new temporal primitives; (2) a set of NL-motivated operators with a granularity-enriched type system facilitates the representation of the *intensional* meaning of a tem-

[1] Project RADAR, http://www.radar.cs.cmu.edu/external.asp

poral expression in a compositional way; and (3) the use of temporal references such as "focus" in the representation cleanly separates the core meaning of an expression from its contextual dependency.

The rest of this paper is organized as follows. Sec. 2 first surveys the characteristics of temporal expressions in emails compared to those in newswire texts, and motivates the design of our representation. Sec 3 then introduces the formalism TCNL. The system TEA and the anchoring process is detailed in Sec. 4, and the evaluation of the system is reported in Sec. 5. Finally Sec. 6 concludes this paper and outlines the future work.

2 Temporal Expressions in Emails

The extent of temporal expressions considered in this paper includes most of the expressions using temporal terms such as *2005*, *summer*, *evening*, *1:30pm*, *tomorrow*, etc. These expressions can be classified into the following categories:

- **Explicit**: These expressions can be immediately anchored, i.e., positioned on a timeline. E.g., *June 2005*, *1998 Summer*, etc.

- **Deictic**: These expressions form a specific relation with the *speech time* (timestamp of an email). E.g., *tomorrow*, *last year*, *two weeks from today*.

- **Relative**: These include the other expressions that form a specific relation with a *temporal focus*, i.e., the implicit time central to the discussion. E.g., *from 5 to 7*, *on Wednesday*, etc. Different from the speech time, a temporal focus can shift freely during the discourse.

- **Durational**: These are the expressions that describe certain length in time. E.g., *for about an hour*, *less than 20 minutes*. This is different from an interval expression where both the starting point and the ending point are given (e.g., *from 5 to 7*). Most durational expressions are used to build more complex expressions, e.g., *for the next 20-30 minutes.*

It is worth emphasizing the crucial difference between deictic expressions and relative expressions: anchoring the former only relies on the fixed speech time while normalizing the latter requires the usually hidden focus. As illustrated below the latter task can be much more challenging:

> "*I'm free next week. Let's meet **on Wednesday***."
> "*Are you free **on Wednesday**?*"

In the first example the "*Wednesday*" denotes a different date since the first sentence sets up a different focus. To make things even more interesting, verbal tense can also play a role, e.g., "*He finished the report* ***on Wednesday***."

There are other types of temporal expressions such as *recurrence* ("*every Tuesday*") and *rate* expressions ("*twice on Wednesday*") that are not supported in our system, although they are planned in our future work (Sec. 6).

To appreciate the different nature of emails as a genre, an interesting observation can be made by comparing the distributions of temporal expressions in emails and in *newswire texts*. The email corpora we used for development and testing were collected from MBA students of Carnegie Mellon University over the year 1997 and 1998. The 277 students, organized in approximately 50 teams of 4 to 6 members, were participating in a 14-week course and running simulated companies in a variety of market scenarios (Kraut et al., 2004). The original dataset, the CSpace email corpus, contains approximately 15,000 emails. We manually picked 1,196 emails that are related to scheduling - these include scheduling meetings, presentations, or general planning for the groups. The emails are then randomly divided into five sets (**email1** to **email5**), and only four of them are used in this work: **email1** was used to establish our baseline, **email2** and **email5** were used for development, and part of **email4** was used for testing. Table 1 shows some basic statistics of these three datasets[2], and an edited sample email is shown in Fig. 1 (names altered). The most apparent difference comparing these emails to newswire texts is in the percentage of explicit expressions occurring in the two different genres. In (Mani et al., 2003) it was reported that the proportion of such expressions is about 25% in the newswire corpus they

[2] The percentages in some rows do not add up to 100% because some expressions like coordination can be classified into more than one type.

Date: **Thu, 11 Sep 1997 00:14:36 -0500**

I have put an outline out in the n10f1 OpReview directory... *(omitted)*

We have very little time for this. Please call me **Thursday night** to get clarification. I will need graphs and prose in files **by Saturday Noon.**

– Mary

ps. Mark and John , I waited **until AFTER midnight** to send this .

Figure 1: A sample email (edited)

used[3]. In contrast, explicit expressions on average only account for around 9.5% in the three email datasets. This is not surprising given that people tend to use under-specified expressions in emails for economic reasons. Another thing to note is that there are roughly the same number of relative expressions and non-relative expressions. Since non-relative expressions (including deictic expressions) can be anchored without tracking the temporal focus over a discourse and therefore can be dealt with in a fairly straightforward way, we may assign 50% as a somewhat generous baseline performance of any anchoring system[4].

Another difference between emails and newswire texts is that the former is a medium for *communication*: an email can be used as a reply, or can be attached within another email, or even be used to address to multiple recipients. All of this complicates a great deal of our task. Other notable differences are that in emails hour ambiguity tend to appear more often ("*I'll be home at 2.*"), and people tend to be more creative when they compose short messages such as using tables (e.g., an entire column of numbers to denote the number of minutes alloted for each presenter), bullet lists, abbreviations, and different month/day formats ("*1/9*" can mean *January 9* or *September 1*), etc. Emails also contain more "human errors" such as misspellings ("*Thusday*" to mean *Thursday*) and confusion about dates (e.g., using "*tomorrow*" when sending emails around midnight), etc. Overall it is very difficult to recover from this type of errors.

3 Representing Times in Natural Language

This section provides a concise overview of TCNL; readers are referred to (Han and Kohlhase, 2003; Han et al., 2006) for more detail.

TCNL has two major components: a constraint-based model for human calendars and a representational language built on top of the model. Different from the other representations such as Zeit-Gram (Stede and Haas, 1998), TOP (Androutsopoulos, 1999), and TimeML/Timex3 (Saurí et al., 2006), the language component of TCNL is essentially "calendar-agnostic" - any *temporal unit* can be plugged in a formula once it is defined in the calendar model, i.e., the calendar model serves as the lexicon for the TCNL language.

Fig. 2 shows a partial model for the Gregorian calendar used in TEA. The entire calendar model is basically a constraint graph with *partial ordering*. The nodes labeled with "`year`" etc. represent temporal units (or variables when viewed as a constraint satisfaction problem (CSP) (Ruttkay, 1998)), and each unit can take on a set of possible values. The undirected edges represent constraints among the units, e.g., the constraint between `month` and `day` mandates that February cannot have more than 29 days. A temporal expression in NL is then viewed as if it assigns values to some of the units, e.g., "*Friday the 13th*" assigns values to only units `dow` (day-of-week) and `day`. An interval-based AC-3 algorithm with a chronological backtracking mechanism is used to derive at the consistent assignments to the other units, therefore allowing us to iterate to any one of the possible *Friday the 13th*.

The ordering among the units is designated by two relations: *measurement* and *periodicity* (arrows in Fig. 2). These relations are essential for supporting various operations provided by the TCNL language such as determining temporal ordering of two time points, performing arithmetic, and changing temporal granularity, etc. For example, to interpret the expression "*early July*", we identify that *July* is a value of unit `month`, and `month` is measured by `day`. We then obtain the size of *July* in terms of `day` (31) and

[3] Using the North American News Corpus.

[4] This is a bit generous since solving simple calendric arithmetics such as anchoring *last summer* still requires a non-trivial modeling of human calendars; see Sec. 3.

Table 1: Basic statistics of the email corpora

	# of emails	**# of tempex**	**explicit**	**deictic**	**relative**	**durational**
email1	253	300	3 (1%)	139 (46.33%)	158 (52.67%)	N/A
email2	253	344	19 (5.5%)	112 (32.6%)	187 (54.4%)	27 (7.8%)
email4 (part.)	149	279	71 (25.4%)	77 (27.6%)	108 (38.7%)	22 (7.9%)
email5	126	213	14 (6.6%)	105 (49.3%)	92 (43.2%)	3 (1.4%)

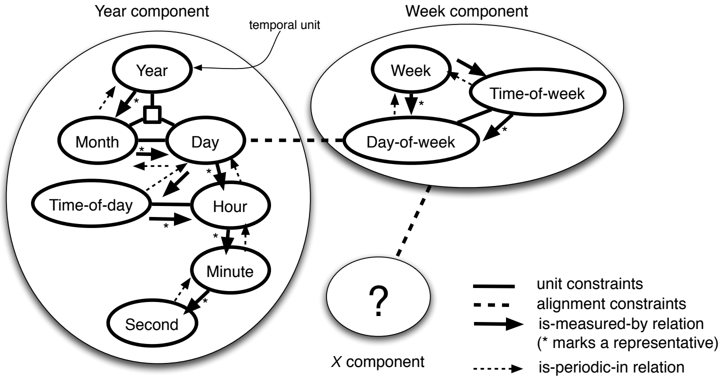

Figure 2: A partial model of the Gregorian calendar

designate the first 10 days (31/3) as the “early” part of July.

Internally the calendar model is further partitioned into several *components*, and different components are *aligned* using non-binary constraints (e.g., in Fig. 2 the `year` component and the `week` component are aligned at the `day` and `dow` units). This is necessary because the top units in these components are not periodic within one another. All of the operations are then extended to deal with multiple calendar components.

Built on top of the calendar model is the *typed* TCNL language. The three major types are *coordinates* (time points; e.g., $\{\texttt{sep},6_{\texttt{day}}\}$ for *September 6*), *quantities* (durations; e.g., $|1_{\texttt{hour}}|$ for *one hour*) and *enumerations* (sets of points, including intervals; e.g., $[\{\texttt{wed}\},\{\texttt{fri}\}]$ for *Wednesday and Friday*). More complex expressions can be represented by using various operators, relations and temporal references; e.g., $\{\texttt{now}-|1_{\texttt{day}}|\}$ for *yesterday*, $\{|1_{\{\texttt{mon}\}}|@\{>= \_\}\}$ for *the coming Monday* (or *the first coming Monday in the future*; the ‘$\_$’ represents the temporal focus), $|<|1_{\texttt{hour}}||$ for *less than one hour*, $[\{\texttt{wed}\}:\{\texttt{fri}\}]$ for *Wednesday to Friday*, $[\texttt{f}\ \{\texttt{sat},\texttt{noon}\}]$ for *by Saturday noon*[5], and $[[\{15_{\texttt{hour}}\}:\{17_{\texttt{hour}}\}]\&\{\texttt{wed}\}]$ for *3-5pm on Wednesday*. The TCNL language is designed in such a way that syntactically different formulae can be evaluated to denote the same date; e.g., $\{\texttt{tue},\texttt{now}+|1_{\texttt{week}}|\}$ (“*Tuesday next week*”) and $\{\texttt{now}+|1_{\{\texttt{tue}\}}|\}$ (“*next Tuesday*”) can denote the same date.

Associated with the operators are type and granularity requirements. For example, when a focus is specified down to *second* granularity, the formula $\{\texttt{now}+|1_{\texttt{day}}|\}$ will return a coordinate at the *day* granularity - essentially stripping away information finer than *day*. This is because the operator ‘+’ (called *fuzzy forward shifting*) requires the left-hand side operand to have the same granularity as that of the right-hand side operand. Type coercion can also happen automatically if it is required by an operator. For example, the operator ‘@’ (*ordinal selection*) requires that the right-hand side operand to be of type enumeration. When presenting a coordinate such as $\{>= \_\}$ (some point in the future), it will be coerced

[5] The `f` denotes the relation “*finishes*” (Allen, 1984); the formula denotes a set of coordinates no later than a Saturday noon.

Table 2: Summary of operators in TCNL; LHS/RHS is the left/right operand, g(e) returns the granularity of e and $\min(s)$ returns the set of minimal units among s.

operator	**Type requirement**	**Granularity requirement**	**Semantics**	**Example**
$+$ and $-$	C × Q → C	g(LHS) ← g(RHS)	fuzzy forward/backward shifting	$\{\texttt{now}+\vert 1_{\texttt{day}}\vert\}$ (“*tomorrow*”)
$++$ and $--$	C × Q → C	g(LHS) ← min(g(LHS)∪g(RHS))	exact forward/backward shifting	$\{\texttt{now}{+}{+}\vert 2_{\texttt{hour}}\vert\}$ (“*2 hours from now*”)
@	Q × E → C	g(RHS) ← g(LHS)	ordinal	$\{\vert 2_{\{\texttt{sun}\}}\vert @\{\texttt{may}\}\}$ (“*the 2nd Sunday in May*”)
&	C × C → C C × E → E E × C → E E × E → E	g(LHS) ← min(g(LHS)∪g(RHS))	distribution	$\{\texttt{now}\ \&\{\texttt{now}+\vert 1_{\texttt{year}}\vert\}\}$ (“*this time next year*”) $[\{\texttt{15}_{\texttt{hour}}\}\&[\{\texttt{wed}\}:\{\texttt{fri}\}]]$ (“*3pm from Wednesday to Friday*”)

into an enumeration so that the ordinal operator can select a requested element out of it. These designs make granularity change and re-interpretation part of a transparent process. Table 2 lists the operators in the TCNL language.

Most of under-specified temporal expressions still lack necessary information in themselves in order to be anchored. For example, it is not clear what to make out of “*on Wednesday*” with no context. In TCNL more information can be supplied by using one of the coordinate *prefixes*: the ‘$+$’/‘$-$’ prefix signifies the relation of a coordinate with the focus (after/before the focus), and the ‘f’/‘p’ indicates the relation of a coordinate with the speech time (future/past). For example, the Wednesday in “*the company will announce on Wednesday*” is represented as $+\texttt{f}\{\texttt{wed}\}$, while “*the company announced on Wednesday*” is represented as $-\texttt{p}\{\texttt{wed}\}$. When evaluating these formulae, TEA will rewrite the former into $\{|1_{\{\texttt{wed}\}}|@\{>=\_,\ >=\texttt{now}\}\}$ and the latter into $\{-|1_{\{\texttt{wed}\}}|@\{<=\_,\ <=\texttt{now}\}\}$ if necessary, essentially trying to find the nearest Wednesday either in the future or in the past. Since TCNL formulae can be embedded, prefixed coordinates can also appear inside a more complex formula; e.g., $\{\{|2_{\{\texttt{sun}\}}|@\texttt{f}\{\texttt{may}\}\}+|2_{\texttt{day}}|\}$ represents “*2 days after a future Mother’s day*”[6].

Note that TCNL itself does not provide a mechanism to instantiate the temporal focus (‘_’). The responsibility of shifting a focus whenever necessary (*focus tracking*) is up to TEA, which is described in the next section.

4 TEA: Temporal Expression Anchorer

The input to our system TEA is English texts with temporal expression markups, and the output is a time string for each temporal expression. The format of a time string is similar to the ISO 8601 scheme: for a time point the format is `YYYYMMDDTHHMMSS` (`T` is a separator), for an interval it is a pair of points separated by ‘/’ (slash). Also whenever there are slots that lack information, we use ‘?’ (question mark) in its place. If a point can reside at any place between two bounds, we use `(lower..upper)` to represent it. Table. 3 shows the TEA output over the example email given in Fig. 1 (`min` and `max` are the minimal and the maximal time points TEA can reason with).

TEA uses the following procedure to anchor each temporal expression:

1. The speech time (variable `now`) and the focus (‘_’) is first assigned to a timestamp (e.g., the received date of an email).

2. For each temporal expression, its nearest verb chunk is identified using the part-of-speech tags of the sentence. Expressions associated with a verb of past tense or present imperfective will be given prefix “$-$p” to its TCNL formula, otherwise it is given “$+$f”[7].

3. A finite-state parser is then used to transduce an expression into its TCNL formula. At the parsing stage the tense and granularity information is available to the parser.

[6] This denotes a possible range of dates, but it is still different from an enumeration.

[7] This is of course a simplification; future work needs to be done to explore other possibilities.

Table 3: Anchoring example for the email in Fig. 1

Expression	**TCNL formula**	**Temporal focus (f)**	**Anchored time string**
(timestamp)			19970911T001436
Thursday night	+f{thu,night}	19970911T001436	(19970911T18????.. 19970911T23????)
by Saturday Noon	[f +f{sat,noon}]	(19970911T18????.. 19970911T23????)	min/19970913T12????
until AFTER midnight	[f{>= −p{midnight}}]	19970913T12????	min/(19970911..max)

4. The produced TCNL formula (or formulae when ambiguity arises) is then evaluated with the speech time and the current focus. In case of ambiguity, one formula will be chosen based on certain heuristics (below). The result of the evaluation is the final output for the expression.

5. *Recency-based* focus tracking: we use the following procedure to determine if the result obtained above can replace the current focus (below). In cases where the result is an ambiguous coordinate (i.e., it denotes a possible range of points), if one of the bounds is `min` or `max`, we use the other to be the new focus; if it is not possible, we choose to keep the focus unchanged. On the other hand, if the result is an enumeration, we go through a similar procedure to avoid using an enumeration with a `min`/`max` bound as the new focus. Finally no quantity can become a focus.

Note that in Step 3 the decision to make partial semantics of a temporal expression available to our parser is based on the following observation: consider the two expressions below

”*Tuesday before Christmas*”
= $\{\texttt{tue}, < \{|25_{\texttt{day}}|@\{\texttt{dec}\}\}\}$

”*Tuesday before 6pm*”
= $\{< \{\texttt{tue},18_{\texttt{hour}}\}, \textbf{de}\ \{\texttt{tue}\}\}$

Both expressions share the same “X before Y” pattern, but their interpretations are different[8]. The key to discriminate the two is to compare the granularities of X and Y: if Y is at a coarser granularity then the first interpretation should be adopted.

In Step 4 we use the following procedure to disambiguate the result:

1. Remove any candidate that resulted in an inconsistency when solving for a solution in the calendar CSP.

2. If the result is meant to be a coordinate, pick the one that is closest to the focus.

3. If the result is supposed to be an enumeration, pick the one whose starting point is closest to the focus, and whose length is the shortest one.

4. Otherwise pick the first one as the result.

For example, if the current time is 2:00 pm, for expression “*at 3*” with a present/future tense, the best answer is 15:00. For expression “*from 3 to 5*”, the best answer is from 3 pm to 5 pm.

When deciding whether a temporal expression can become the next focus, we use simple heuristics to rule out any expression that behaves like a noun modifier. This is motivated by the following example (timestamp: `19970919T103315`):

> *IT basically analyses the breakdown on labor costs and compares our* ***1998*** *labor costs with their demands for* ***1999-2000****.*
>
> ...
>
> *I will check mail* ***on Sunday*** *and see any feedback.*

Without blocking the expression *1999-2000* from becoming the focus, the last expression will be incorrectly anchored in year 2000. The key observation here is that a noun-modifying temporal expression usually serves as a temporal co-reference instead of representing a new temporal entity in the discourse. These references tend to have a more confined effect in anchoring the subsequent expressions.

[8] **de** denotes a relation “during or equal” (Allen, 1984).

Table 4: Development and testing results

	Accuracy	Parsing errors	Human errors	Anchoring errors
email2 (dev)	78.2%	10.47%	1.7%	9.63%
email5 (dev)	85.45%	5.16%	1%	8.39%
email4 (testing)	76.34%	17.92%	$<1\%$	5.74%

5 Evaluation

The temporal expressions in all of the datasets were initially tagged using rules developed for MinorThird[9], and subsequently corrected manually by two of the authors. We then developed a prototype system and established our baseline over **email1** (50%). The system at that time did not have any focus tracking mechanism (i.e., it always used the timestamp as the focus), and it did not use any tense information. The result confirms our estimate given in Sec. 2. We then gradually developed TEA to its current form using **email1**, **email2** and **email5**. During the four-month development we added the focus tracking mechanism, incorporating the tense information into each TCNL formula via the coordinate prefixes, and introduced several representational improvements. Finally we tested the system on the unseen dataset **email4**, and obtained the results shown in Table 4. Note that the percentages reported in the table are *accuracies*, i.e., the number of correctly anchored expressions over the total number of temporal expressions over a dataset, since we are assuming correct tagging of all of the expressions. Our best result was achieved in the dev set **email5** (85.45%), and the accuracy over the test set **email4** was 76.34%.

Table 4 also lists the types of the errors made by our system. The parsing errors are mistakes made at transducing temporal expressions using the finite-state parser into their TCNL formulae, the human errors are described in Sec. 2, and the rest are the anchoring errors. The accuracy numbers are all compared favorably to the baseline (50%). To put this performance in perspective, in (Wiebe et al., 1998) a similar task was performed over transcribed scheduling-related phone conversations. They reported an average accuracy 80.9% over the CMU test set and 68.9% over the NMSU test set. Although strictly speaking the two results cannot be compared due to differences in the nature of the corpora (transcription vs. typing), we nevertheless believe it represents a closer match compared to the other works done on newswire genre.

It should also be noted that we adopted a similar recency-based focus model as in (Wiebe et al., 1998). Although simple to implement, this naive approach proved to be one major contributor to the anchoring errors in our experiments. An example is given below (the anchored times are shown in subscript):

> *This research can not proceed until the trade-offs are known **on Monday***$_{19970818}$.
>
> ...
>
> *Mary will perform this **by Friday***$_{(min..19970822)}$ *using the data **from Monday***$_{19970825}$.

The last expression received an incorrect date: it should be the same date the expression "*on Monday*" refers to. Our system made this error because it blindly used the most recently mentioned time (`(min..19970822)`) as the focus to anchor the formula $+$`f{mon}`. This error later also propagated to the anchoring of the subsequent expressions.

6 Conclusion and Future Work

In this paper we have adopted a constraint-based representation of time, *Time Calculus for Natural Language* (TCNL), to tackle the task of anchoring temporal expressions in a novel genre, *emails*. We believe that the genre is sufficiently different from newswire texts, and its highly under-specified nature fits well with a constraint-based modeling of human calendars. TCNL also allows for an explicit representation of *temporal focus*, and many of our intuitions about *granularity change* and *temporal arithe-*

[9] `http://minorthird.sourceforge.net/`

matics are encapsulated in its type system and operators. The performance of our anchoring system is significantly better than baseline, and compares favorably with some of the closely related work.

In the future we will re-examine our focus tracking mechanism (being the most significant source of errors), and possibly treat it as a classification problem (similar to (Mani et al., 2003)). We also need to investigate the disambiguation procedure and possibly migrate the functionality into a separate discourse module. In addition, the co-referencing tendency of noun-modifying expressions could lead to a better way to anchoring this particular type of temporal expressions. Finally we would like to expand our coverage of temporal expressions to include other types of expressions such as recurrence expressions[10].

Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA), or the Department of Interior-National Business Center (DOI-NBC).

References

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

I. Androutsopoulos. 1999. Temporal Meaning Representations in a Natural Language Front-end. In M. Gergatsoulis and P. Rondogiannis, editors, *Intensional Programming II (Proceedings of the 12th International Symposium on Languages for Intensional Programming*, Athens, Greece.

E. Filatova and E. Hovy. 2001. Assigning Time-Stamps To Event-Clauses. In *Proceedings of ACL-2001: Workshop on Temporal and Spatial Information Processing*, Toulouse, France, 7.

Benjamin Han and Michael Kohlhase. 2003. A Time Calculus for Natural Language. In *The 4th Workshop on Inference in Computational Semantics*, Nancy, France, September.

B. Han, D. Gates, and L. Levin. 2006. From Language to Time: A Temporal Expression Anchorer. In *Proceedings of the 13th International Symposium on Temporal Representation and Reasoning (TIME 2006)*, Budapest, Hungary.

J. R. Hobbs and Feng. Pan. 2004. An ontology of time for the semantic web. *TALIP Special Issue on Spatial and Temporal Information Processing*, 3(1):66–85, March.

R. E. Kraut, S. R. Fussell, F. J. Lerch, and A Espinosa. 2004. Coordination in teams: Evidence from a simulated management game. *Journal of Organizational Behavior*, to appear.

I. Mani and G. Wilson. 2000. Robust Temporal Processing of News. In *Proceedings of ACL-2000.*

I. Mani, B. Schiffman, and J. Zhang. 2003. Inferring Temporal Ordering of Events in News. In *Proceedings of the Human Language Technology Conference (HLT-NAACL'03).*

Zsófia Ruttkay. 1998. Constraint Satisfaction - a Survey. Technical Report 11(2-3), CWI.

Roser Saurí, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky, 2006. *TimeML Annotation Guidelines, Version 1.2.1*, January 31.

F. Schilder and C. Habel. 2001. From Temporal Expressions To Temporal Information: Semantic Tagging Of News Messages. In *Proceedings of ACL-2001: Workshop on Temporal and Spatial Information Processing*, Toulouse, France, 7.

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

M. Stede and S. Haas. 1998. Understanding and tracking temporal descriptions in dialogue. In B. Schröder, W. Lenders, W. Hess, and T. Portele, editors, *Proceedings of the 4th Conference on Natural Language Processing - KONVENS '98.*

N. Vazov. 2001. A System for Extraction of Temporal Expressions from French Texts Based on Syntactic and Semantic Constraints. In *Proceedings of ACL-2001: Workshop on Temporal and Spatial Information Processing*, Toulouse, France, 7.

J. M. Wiebe, T. P. O'Hara, T. Ohrstrom-Sandgren, and K. J. McKeever. 1998. An Empirical Approach to Temporal Reference Resolution. *Journal of Artificial Intelligence Research*, 9:247–293.

[10] The current design of TCNL allows for a more restricted type of recurrence: e.g., "*3pm from Wednesday to Friday*" is represented as $[\{\texttt{15}_{\texttt{hour}}\}\&[\{\texttt{wed}\}:\{\texttt{fri}\}]]$. However this is insufficient to represent expressions such as "*every 4 years*".