Robust Temporal Processing of News

Inderjeet Mani and George Wilson

The MITRE Corporation, W640

11493 Sunset Hills Road

Reston, Virginia 22090

{imani, gwilson}@mitre.org

Abstract

We introduce an annotation scheme for temporal expressions, and describe a method for resolving temporal expressions in print and broadcast news. The system, which is based on both hand-crafted and machine-learnt rules, achieves an 83.2% accuracy (F-measure) against hand-annotated data. Some initial steps towards tagging event chronologies are also described.

Introduction

The extraction of temporal information from news offers many interesting linguistic challenges in the coverage and representation of temporal expressions. It is also of considerable practical importance in a variety of current applications. For example, in question-answering, it is useful to be able to resolve the underlined reference in "the next year, he won the Open" in response to a question like "When did X win the U.S. Open?". In multi-document summarization, providing fine-grained chronologies of events over time (e.g., for a biography of a person, or a history of a crisis) can be very useful. In information retrieval, being able to index broadcast news stories by event times allows for powerful multimedia browsing capabilities.

Our focus here, in contrast to previous work such as (MUC 1998), is on *resolving* time expressions, especially indexical expressions like "now", "today", "tomorrow", "next Tuesday", "two weeks ago", "20 mins after the next hour", etc., *which designate times that are dependent on the speaker and some*

"reference" time[1]. In this paper, we discuss a temporal annotation scheme for representing dates and times in temporal expressions. This is followed by details and performance measures for a tagger to extract this information from news sources. The tagger uses a variety of hand-crafted and machine-discovered rules, all of which rely on lexical features that are easily recognized. We also report on a preliminary effort towards constructing event chronologies from this data.

1 Annotation Scheme

Any annotation scheme should aim to be simple enough to be executed by humans, and yet precise enough for use in various natural language processing tasks. Our approach (Wilson et al. 2000) has been to *annotate those things that a human could be expected to tag.*

Our representation of times uses the ISO standard CC:YY:MM:DD:HH:XX:SS, with an optional time zone (ISO-8601 1997). In other words, time points are represented in terms of a calendric coordinate system, rather than a real number line. The standard also supports the representation of weeks and days of the week in the format CC:YY:Wwwd where ww specifies which week within the year (1-53) and d specifies the day of the week (1-7). For example, "last week" might receive the VAL 20:00:W16. A time (TIMEX) expression (of type TIME or DATE) representing a particular point on the ISO line, e.g., "Tuesday, November 2, 2000" (or "next Tuesday") is represented with the ISO time Value (VAL), 20:00:11:02. Interval expressions like "From

---

[1] Some of these indexicals have been called "relative times" in the (MUC 1998) temporal tagging task.

May 1999 to June 1999", or "from 3 pm to 6 pm" are represented as two separate TIMEX expressions.

In addition to the values provided by the ISO standard, we have added several extensions, including a list of additional tokens to represent some commonly occurring temporal units; for example, "summer of '69" could be represented as 19:69:SU. The intention here is to capture the information in the text while leaving further interpretation of the Values to applications using the markup.

It is worth noting that there are several kinds of temporal expressions that are not to be tagged, and that other expressions tagged as a time expression are not assigned a value, because doing so would violate the simplicity and preciseness requirements. We do not tag unanchored intervals, such as "half an hour (long)" or "(for) one month". Non-specific time expressions like generics, e.g., "April" in "April is usually wet", or "today" in "today's youth", and indefinites, e.g., "a Tuesday", are tagged without a value. Finally, expressions which are ambiguous without a strongly preferred reading are left without a value.

This representation treats points as primitive (as do (Bennett and Partee 1972), (Dowty 1979), among others); other representations treat intervals as primitive, e.g., (Allen 1983). Arguments can be made for either position, as long as both intervals and points are accommodated. The annotation scheme does not force committing to end-points of intervals, and is compatible with current temporal ontologies such as (KSL-Time 1999); this may help eventually support advanced inferential capabilities based on temporal information extraction.

2 Tagging Method

Overall Architecture

The system architecture of the temporal tagger is shown in Figure 1. The tagging program takes in a document which has been tokenized into words and sentences and tagged for part-of-speech. The program passes each sentence first to a module that identifies time expressions, and then to another module (SC) that resolves self-contained time expressions. The program then takes the entire document and passes it to a discourse processing module (DP) which resolves context-dependent time expressions (indexicals as well as other expressions). The DP module tracks transitions in temporal focus, uses syntactic clues, and various other knowledge sources. The module uses a notion of *Reference Time* to help resolve context-dependent expressions. Here, the *Reference Time* is the time a context-dependent expression is relative to. In our work, the reference time is assigned the value of either the *Temporal Focus* or the document (creation) date. The *Temporal Focus* is the time currently being talked about in the narrative. The initial reference time is the document date.

2.2 Assignment of time values

We now discuss the modules that assign values to identified time expressions. Times which are fully specified are tagged with their value, e.g, "June 1999" as 19:99:06 by the SC module. The DP module uses an ordered sequence of rules to handle the context-dependent expressions. These cover the following cases:

Explicit offsets from reference time: indexicals like "yesterday", "today", "tomorrow", "this afternoon", etc., are ambiguous between a specific and a non-specific reading. The specific use (distinguished from the generic one by machine learned rules discussed below) gets assigned a value based on an offset from the reference time, but the generic use does not.

Positional offsets from reference time: Expressions like "next month", "last year" and "this coming Thursday" use lexical markers (underlined) to describe the direction and magnitude of the offset from the reference time.

Implicit offsets based on verb tense: Expressions like "Thursday" in "the action taken Thursday", or bare month names like "February" are passed to rules that try to determine the direction of the offset from

the reference time. Once the direction is determined, the magnitude of the offset can be computed. The tense of a neighboring verb is used to decide what direction to look to resolve the expression. Such a verb is found by first searching backward to the last TIMEX, if any, in the sentence, then forward to the end of the sentence and finally backwards to the beginning of the sentence. If the tense is past, then the direction is backwards from the reference time. If the tense is future, the direction is forward. If the verb is present tense, the expression is passed on to subsequent rules for resolution. For example, in the following passage, "Thursday" is resolved to the Thursday prior to the reference date because "was", which has a past tense tag, is found earlier in the sentence:

> The Iraqi news agency said the first shipment of 600,000 barrels <u>was</u> loaded *Thursday* by the oil tanker Edinburgh.

Further use of lexical markers: Other expressions lacking a value are examined for the nearby presence of a few additional markers, such as "since" and "until", that suggest the direction of the offset.

Nearby Dates: If a direction from the reference time has not been determined, some dates, like "Feb. 14", and other expressions that indicate a particular date, like "Valentine's Day", may still be untagged because the year has not been determined. If the year can be chosen in a way that makes the date in question less than a month from the reference date, that year is chosen. For example, if the reference date is Feb. 20, 2000 and the expression "Feb. 14" has not been assigned a value, this rule would assign it the value Feb. 14, 2000. Dates more than a month away are not assigned values by this rule.

3 Time Tagging Performance

3.1 Test Corpus

There were two different genres used in the testing: print news and broadcast news transcripts. The print news consisted of 22 New York Times (NYT) articles from January 1998. The broadcast news data consisted of 199 transcripts of Voice of America (VOA) broadcasts from January of 1998, taken from the TDT2 collection (TDT2 1999).   The print data was much cleaner than the transcribed broadcast data in the sense that there were very few typographical errors, spelling and grammar were good. On the other hand, the print data also had longer, more complex sentences with somewhat greater variety in the words used to represent dates. The broadcast collection had a greater proportion of expressions referring to time of day, primarily due to repeated announcements of the current time and the time of upcoming shows.

The test data was marked by hand tagging the time expressions and assigning value to them where appropriate. This hand-marked data was used to evaluate the performance of a frozen version of the machine tagger, which was trained and engineered on a separate body of NYT, ABC News, and CNN data. Only the body of the text was included in the tagging and evaluation.

3.2 System performance

The system performance is shown in Table 1[2]. Note that if the human said the TIMEX had no value, and the system decided it had a value, this is treated as an error. A baseline of just tagging values of absolute, fully specified TIMEXs (e.g., "January 31st, 1999") is shown for comparison in parentheses. Obviously, we would prefer a larger data sample; we are currently engaged in an effort within the information extraction community to annotate a large sample of the TDT2 collection and to conduct an inter-annotator reliability study.

Error Analysis

Table 2 shows the number of errors made by the program classified by the type of error. Only 2 of these 138 errors (5 on TIME, 133 on DATE) were due to errors in the source. 14 of the 138 errors (9 NYT vs. 5 VOA)

---

were due to the document date being incorrect as a reference time.

Part of speech tagging: Some errors, both in the identification of time expressions and the assignment of values, can be traced to incorrect part of speech tagging in the preprocessing; many of these errors should be easily correctable.

TIMEX expressions

A total of 44 errors were made in the identification of TIMEX expressions.

Not yet implemented: The biggest source of errors in identifying time expressions was formats that had not yet been implemented. For example, one third (7 of 21, 5 of which were of type TIME) of all missed time expressions came from numeric expressions being spelled out, e.g. "nineteen seventy-nine". More than two thirds (11 of 16) of the time expressions for which the program incorrectly found the boundaries of the expression (bad extent) were due to the unimplemented pattern "Friday the 13th". Generalization of the existing patterns should correct these errors.

Proper Name Recognition: A few items were spuriously tagged as time expressions (extra TIMEX). One source of this that should be at least partially correctable is in the tagging of apparent dates in proper names, e.g. "The July 26 Movement", "The Tonight Show", "USA Today". The time expression identifying rules assumed that these had been tagged as lexical items, but this lexicalization has not yet been implemented.

Values assigned

A total of 94 errors were made in the assignment of values to time expressions that had been correctly identified.

Generic/Specific: In the combined data, 25 expressions were assigned a value when they should have received none because the expression was a generic usage that could not be placed on a time line. This is the single biggest source of errors in the value assignments.

4 Machine Learning Rules

Our approach has been to develop initial rules by hand, conduct an initial evaluation on an unseen test set, determine major errors, and then handling those errors by augmenting the rule set with additional rules discovered by machine learning. As noted earlier, distinguishing between specific use of a time expression and a generic use (e.g., "today", "now", etc.) was and is a significant source of error. Some of the other problems that these methods could be applied to distinguishing a calendar year reference from a fiscal year one (as in "this year"), and distinguishing seasonal from specific day references. For example, "Christmas" has a seasonal use (e.g., "I spent Christmas visiting European capitals") distinct from its reference to a specific day use as "December 25th" (e.g., "We went to a great party on Christmas").

Here we discuss machine learning results in distinguishing specific use of "today" (meaning the day of the utterance) from its generic use meaning "nowadays". In addition to features based on words co-occurring with "today" (*Said*, *Will*, *Even*, *Most*, and *Some* features below), some other features (*DOW* and *CCYY*) were added based on a granularity hypothesis. Specifically, it seems possible that "today" meaning the day of the utterance sets a scale of events at a day or a small number of days. The generic use, "nowadays", seems to have a broader scale. Therefore, terms that might point to one of these scales such as the names of days of the week, the word "year" and four digit years were also included in the training features. To summarize, the features we used for the "today" problem are as follows (features are boolean except for string-valued *POS1* and *POS2*):

Poss: whether "today" has a possessive inflection
Qcontext: whether "today" is inside a quotation
Said: presence of "said" in the same sentence
Will: presence of "will" in the same sentence
Even: presence of "even" in the same sentence
Most: presence of "most" in the same sentence
Some: presence of "some" in the same

sentence
Year: presence of "year" in the same sentence
CCYY: presence of a four-digit year in the same sentence
DOW: presence of a day of the week expression ("Monday" thru "Sunday") in the same sentence
FW: "today" is the first word of the sentence
POS1: part-of-speech of the word before "today"
POS2: part-of-speech of the word after "today"
Label: specific or non-specific (class label)

Table 3 shows the performance of different classifiers in classifying occurrences of "today" as generic versus specific. The results are for 377 training vectors and 191 test vectors, measured in terms of Predictive Accuracy (percentage test vectors correctly classified).

We incorporated some of the rules learnt by C4.5 Rules (the only classifier which directly output rules) into the current version of the program. These rules included classifying "today" as generic based on (1) feature *Most* being true (74.1% accuracy) or (2) based on feature *FW* being true and *Poss*, *Some* and *Most* being false (67.4% accuracy). The granularity hypothesis was partly borne out in that C4.5 rules also discovered that the mention of a day of a week (e.g. "Monday"), anywhere in the sentence predicted specific use (73.3% accuracy).

5 Towards Chronology Extraction

Event Ordering
Our work in this area is highly preliminary. To extract temporal relations between events, we have developed an event-ordering component, following (Song and Cohen 1991). We encode the tense associated with each verb using their modified Reichenbachian (Reichenbach 1947) representation based on the tuple $<s_i,\ lge,\ r_i,\ lge,\ e_i>$. Here s_i is an index for the speech time, r_i for the reference time, and e_i for the event time, with *lge* being the temporal relations *precedes*, *follows*, or *coincides*. With each successive event, the temporal focus is either maintained or shifted, and a temporal ordering relation between the event and the focus is asserted, using heuristics defining coherent tense sequences; see (Song and Cohen 1991) for more details. Note that the tagged TIME expressions aren't used in determining these inter-event temporal relations, so this event-ordering component could be used to order events which don't have time VALs.

Event Time Alignment
In addition, we have also investigated the alignment of events on a calendric line, using the tagged TIME expressions. The processing, applied to documents tagged by the time tagger, is in two stages. In the first stage, for each sentence, each "taggable verb occurrence" lacking a time expression is given the VAL of the immediately previous time expression in the sentence. Taggable verb occurrences are all verb occurrences except auxiliaries, modals and verbs following "to", "not", or specific modal verbs. In turn, when a time expression is found, the immediately previous verb lacking a time expression is given that expression's VAL as its TIME. In the second stage, each taggable verb in a sentence lacking a time expression is given the TIME of the immediately previous verb in the sentence which has one, under the default assumption that the temporal focus is maintained.

Of course, rather than blindly propagating time expressions to events based on proximity, we should try to represent relationships expressed by temporal coordinators like "when", "since", "before", as well as explicitly temporally anchored events, like "ate at 3 pm". The event-aligner component uses a very simple method, intended to serve as a baseline method, and to gain an understanding of the issues involved. In the future, we expect to advance to event-alignment algorithms which rely on a syntactic analysis, which will be compared against this baseline.

Assessment
An example of the chronological tagging of events offered by these two components is shown in Figure 2, along with the TIMEX tags extracted by the time tagger. Here each

taggable verb is given an event index, with the *precedes* attribute indicating one or more event indices which it precedes temporally. (Attributes irrelevant to the example aren't shown). The information of the sort shown in Figure 2 can be used to sort and cluster events temporally, allowing for various time-line based presentations of this information in response to specific queries.

The event-orderer has not yet been evaluated. Our evaluation of the event-aligner checks the TIME of all correctly recognized verbs (i.e., verbs recognized correctly by the part-of-speech tagger). The basic criterion for event TIME annotation is that if the time of the event is obvious, it is to be tagged as the TIME for that verb. (This criterion excludes interval specifications for events, as well as event references involving generics, counterfactuals, etc. However, the judgements are still delicate in certain cases.) We score Correctness as *number of correct TIME fills for correctly recognized verbs over total number of correctly recognized verbs*. Our total correctness scores on a small sample of 8505 words of text is 394 correct event times out of 663 correct verb tags, giving a correctness score of 59.4%. Over half the errors were due to propagation of spreading of an incorrect event time to neighboring events; about 15% of the errors were due to event times preceding the initial TIMEX expression (here the initial reference time should have been used); and at least 10% of the errors were due to explicitly marked tense switches. This is a very small sample, so the results are meant to be illustrative of the scope and limitations of this baseline event-aligning technique rather than present a definitive result.

6 Related Work

The most relevant prior work is (Wiebe et al. 98), who dealt with meeting scheduling dialogs (see also (Alexandersson et al. 97), (Busemann et al. 97)), where the goal is to schedule a time for the meeting. The temporal references in meeting scheduling are somewhat more constrained than in news, where (e.g., in a historical news piece on toxic dumping) dates and times may be relatively unconstrained. In addition, their model requires the maintenance of a focus stack. They obtained roughly .91 Precision and .80 Recall on one test set, and .87 Precision and .68 Recall on another. However, they adjust the reference time during processing, which is something that we have not yet addressed.

More recently, (Setzer and Gaizauskas 2000) have independently developed an annotation scheme which represents both time values and more fine-grained inter-event and event-time temporal relations. Although our work is much more limited in scope, and doesn't exploit the internal structure of events, their annotation scheme may be leveraged in evaluating aspects of our work.

The MUC-7 task (MUC-7 98) did not require VALs, but did test TIMEX recognition accuracy. Our 98 F-measure on NYT can be compared for just TIMEX with MUC-7 (MUC-7 1998) results on similar news stories, where the best performance was .99 Precision and .88 Recall. (The MUC task required recognizing a wider variety of TIMEXs, including event-dependent ones. However, at least 30% of the dates and times in the MUC test were fixed-format ones occurring in document headers, trailers, and copyright notices.)

Finally, there is a large body of work, e.g., (Moens and Steedman 1988), (Passoneau 1988), (Webber 1988), (Hwang 1992), (Song and Cohen 1991), that has focused on a computational analysis of tense and aspect. While the work on event chronologies is based on some of the notions developed in that body of work, we hope to further exploit insights from previous work.

Conclusion

We have developed a temporal annotation specification, and an algorithm for resolving a class of time expressions found in news. The algorithm, which is relatively knowledge-poor, uses a mix of hand-crafted

and machine-learnt rules and obtains reasonable results.

In the future, we expect to improve the integration of various modules, including tracking the temporal focus in the time resolver, and interaction between the event-order and the event-aligner. We also hope to handle a wider class of time expressions, as well as further improve our extraction and evaluation of event chronologies. In the long run, this could include representing event-time and inter-event relations expressed by temporal coordinators, explicitly temporally anchored events, and nominalizations.
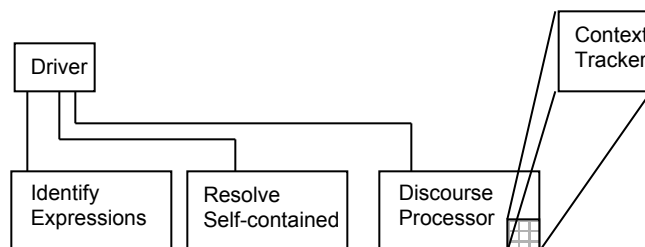

Figure 1. Time Tagger

Source articles number of words	Type	Human Found (Correct)	System Found	System Correct	Precision	Recall	F-measure
NYT 22 35,555	TIMEX	302	302	296	98.0	98.0	98.0
	Values	302	302	249 (129)	82.5 (42.7)	82.5 (42.7)	82.5 (42.7)
Broadcast 199 42,616	TIMEX	426	417	400	95.9	93.9	94.9
	Values	426	417	353 (105)	84.7 (25.1)	82.9 (24.6)	83.8 (24.8)
Overall 221 78,171	TIMEX	728	719	696	96.8	95.6	96.2
	Values	728	719	602 (234)	83.7 (32.5)	82.7 (32.1)	83.2 (32.3)

Table 1. Performance of Time Tagging Algorithm

	Print	Broadcast	Total
Missing Vals	10	29	39
Extra Vals	18	7	25
Wrong Vals	19	11	30
Missing TIMEX	6	15	21
Extra TIMEX	2	5	7
Bad TIMEX extent	4	12	16
TOTAL	59	79	138

Table 2. High Level Analysis of Errors

Algorithm	Predictive Accuracy
MC4 Decision Tree[3]	79.8
C4.5 Rules	69.8
Naïve Bayes	69.6
Majority Class (specific)	66.5

Table 3. Performance of "Today" Classifiers

In the last step after years of preparation, the countries <lex eindex="9" precedes="10|" TIME="19981231">locked</lex> in the exchange rates of their individual currencies to the euro, thereby <lex eindex="10" TIME="19981231">setting</lex> the value at which the euro will begin <lex eindex="11" TIME="19990104">trading</lex> when financial markets open around the world on <TIMEX VAL="19990104">Monday</TIMEX>…….

Figure 2. Chronological Tagging

References

J. Alexandersson, N. Riethinger, and E. Maier. *Insights into the Dialogue Processing of VERBMOBIL*. Proceedings of the Fifth Conference on Applied Natural Language Processing, 1997, 33-40.

J. F. Allen. *Maintaining Knowledge About Temporal Intervals*. Communications of the ACM, Volume 26, Number 11, 1983.

M. Bennett and B. H. Partee. *Towards the Logic of Tense and Aspect in English*, Indiana University Linguistics Club, 1972.

S. Busemann, T. Decleck, A. K. Diagne, L. Dini, J. Klein, and S. Schmeier. *Natural Language Dialogue Service for Appointment Scheduling Agents*. Proceedings of the Fifth Conference on Applied Natural Language Processing, 1997, 25-32.

D. Dowty. "Word Meaning and Montague Grammar", D. Reidel, Boston, 1979.

C. H. Hwang. A Logical Approach to Narrative Understanding. Ph.D. Dissertation, Department of Computer Science, U. of Alberta, 1992.

ISO-8601 ftp://ftp.qsl.net/pub/g1smd/8601v03.pdf 1997.

R. Kohavy and D. Sommerfield. MLC++: Machine Learning Library in C++. http://www.sgi.com/Technology/mlc 1996.

KSL-Time 1999. http://www.ksl.Stanford.EDU/ontologies/time/ 1999.

M. Moens and M. Steedman. *Temporal Ontology and Temporal Reference*. Computational Linguistics, 14, 2, 1988, pp. 15-28.

MUC-7. Proceedings of the Seventh Message Understanding Conference, DARPA. 1998.

R. J. Passonneau. *A Computational Model of the Semantics of Tense and Aspect*. Computational Linguistics, 14, 2, 1988, pp. 44-60.

H. Reichenbach. Elements of Symbolic Logic. London, Macmillan. 1947.

A. Setzer and R. Gaizauskas. *Annotating Events and Temporal Information in Newswire Texts*. Proceedings of the Second International Conference On Language Resources And Evaluation (LREC-2000), Athens, Greece, 31 May- 2 June 2000.

F. Song and R. Cohen. *Tense Interpretation in the Context of Narrative*. Proceedings of the Ninth National Conference on Artifical Intelligence (AAAI'91), pp.131-136. 1991.

TDT2 http://morph.ldc.upenn.edu/Catalog/LDC99T37.html 1999

B. Webber. *Tense as Discourse Anaphor*. Computational Linguistics, 14, 2, 1988, pp. 61-73.

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

G. Wilson, I. Mani, B. Sundheim, and L. Ferro. *Some Conventions for Temporal Annotation of Text*. Technical Note (in preparation). The MITRE Corporation, 2000.

---

[3] Algorithm from the MLC++ package (Kohavi and Sommerfield 1996).