Computer Science Department, Brandeis University
Technical Report CS-07-268

Three Approaches to Learning TLINKs in TimeML

Inderjeet Mani[¥¶], Ben Wellner[¥¶], Marc Verhagen[¶] and James Pustejovsky[¶]
[¥]The MITRE Corporation
202 Burlington Road, Bedford, MA 01730, USA
[¶]Department of Computer Science, Brandeis University
415 South St., Waltham, MA 02254, USA
{imani, wellner}@mitre.org, {marc, jamesp}@cs.brandeis.edu

1 Introduction

TimeML (Pustejovsky et al. 2005) (www.timeml.org) is an annotation scheme for markup of events, times, and their temporal relations in news articles. There has been a considerable amount of research activity related to this scheme. In this paper, we focus on the problem of learning temporal relations (called TLINKs) in TimeML.

We draw attention to two problems found in earlier work by (Mani et al. 2006). The first problem is a bug in vector generation (duplicate vectors were included). The second problem is the fact that the evaluation scheme was somewhat unrealistic; simply evaluating on a per-relation basis across relations drawn from all documents doesn't tell us how well we're doing on each document. This paper addresses these two problems.

2 TimeML

The TimeML scheme flags tensed verbs, adjectives, and nominals with EVENT tags with various attributes, including the class of event, tense, grammatical aspect, polarity (negative or positive), any modal operators which govern the event being tagged, and cardinality of the event if it's mentioned more than once. Likewise, time expressions are flagged and their values normalized, based on TIMEX3, an extension of the ACE (2004) (tern.mitre.org) TIMEX2 annotation scheme.

For temporal relations, TimeML defines a TLINK tag that links tagged events to other events and/or times. For example, in (1), a TLINK tag **orders** an instance of the event of entering to an instance of the drinking with the relation type AFTER[1].

```
  (1) Max <EVENT eventID="e1"
class="occurrence" tense="past" as-
pect="none">entered</EVENT> the room.
He <EVENT eventID="e2"
class="occurrence" tense="past" as-
pect="perfect">had drunk</EVENT>a
lot of wine.
  <TLINK eventID="e1" relatedToEven-
tID="e2" relType="AFTER"/>
```

Likewise, in (2), a TLINK tag will **anchor** the event instance of announcing to the time expression *Tuesday* (whose normalized value is inferred from context), with the relation IS_INCLUDED..

```
   (2) The company <EVENT even-
tID="e1" class="reporting"
tense="past" aspect="none" >an-
nounced</EVENT> the results on
<TIMEX3 tid="t2" type="DATE" tempo-
ralFunction="false" value="1998-01-
08">Tuesday </TIMEX3>.
  <TLINK eventID="e1" related-
ToTimeID="t2"
relType="IS_INCLUDED"/>
```

The anchor relation is an Event-Time TLINK, and the order relation is an Event-Event TLINK. TimeML uses 14 temporal relations, which reduce to a disjunctive classification of 6 temporal relations *RelTypes* = {SIMULTANEOUS, IBEFORE, BEFORE, BEGINS, ENDS, INCLUDES}. An event or time is SIMULTANEOUS with another event or time if they occupy the same time interval. An event or time INCLUDES another event or time if the latter occupies a proper subinterval of the former. These 6 relations and their inverses map one-to-one to 12 of Allen's 13 basic relations (Allen 1984).

Of the 14 TLINK relations, the 6 inverse relations are redundant. In order to have a disjunctive classification, SIMULTANEOUS and IDENTITY are collapsed, since IDENTITY is a subtype of SIMULTANEOUS[2]. DURING and IS_INCLUDED are collapsed since DURING is

[1]XML tags are shown in an abbreviated form.

[2]Specifically, X and Y are identical if they are simultaneous *and* coreferential.

a subtype of IS_INCLUDED that anchors events to times that are durations. IBEFORE (immediately before) corresponds to Allen's MEETS. Allen's OVERLAPS relation is not represented in TimeML. More details can be found at timeml.org.

3 Challenges

The annotation of TimeML information is on a par with other challenging semantic annotation schemes, like Coreference,Word Sense Disambiguation, Rhetorical Structure Theory annotation, etc., where high inter-annotator reliability is crucial but not always achievable without massive preprocessing to reduce the user's workload. In TimeML, inter-annotator agreement for time expressions and events is 0.83 and 0.78 F-measure respectively, but on TLINKs it is 0.55 F-measure, due to the large number of event pairs that can be selected for comparison.

Two corpora have been released based on TimeML: the TimeBank (www.timeml.org) (we used version 1.2.a) with 186 documents and 64,077 words of text, and the AQUAINT Corpus (www.timeml.org), with 73 documents and 38,709 words. The TimeBank was developed in the early stages of TimeML development, and was partitioned across five annotators with different levels of expertise. The AQUAINT Corpus was developed recently, and was partitioned across three highly trained annotators. In our experiments, we merged the two datasets to produce a single 259-document corpus, called ATC.

Table 1 shows the number of EVENTs and TIMES, and the distribution of TLINK RelTypes in the ATC[3]. The majority class percentages are shown in parentheses. It can be seen that BEFORE and SIMULTANEOUS together form a majority of event-ordering (Event-Event) links, whereas most of the event anchoring (Event-Time) links are INCLUDES.

4 Framing the Problem

There are several sub-problems related to inferring event anchoring and event ordering. Once a tagger has tagged the events and times, the first task (A) is to link events and/or times, and the second task (B) is to label the links. Task A is difficult to evaluate since, in the absence of massive preprocessing, many links are ignored by the human in creating the annotated corpora. In addition, a program, as a baseline, can trivially link all tagged events and times, getting 100% recall on Task A. We focus here on Task B, the labeling task. In the case of humans, when a TLINK is posited by both annotators between the same pairs of events or times, the inter-annotator agreement on the labels is a .77 F-measure.

Thus, we can consider TLINK labeling as the following classification problem: given an ordered pair of elements X and Y, where X and Y are events or times which the human has related temporally via a TLINK, the classifier has to assign a label in *RelType*s. Using *RelType*s instead of RelTypes $\cup$ NONE also avoids the problem of heavily skewing the data towards the NONE class.

To construct feature vectors for machine learning, we took each TLINK in the corpus and used the given TimeML features, with the TLINK class being the vector's class label. The vectors we used are available at timeml.org.

For learning, we used an off-the-shelf Maximum Entropy (ME) classifier from the Conditional Random Field toolkit Carafe[4].

5 Approach I: Partitioning By Instances

5.1 Introduction

In our earlier experiments (Mani et al., 2006), we took all the TLINKs in the entire ATC corpus, and evaluated the ME classifier using ten-fold cross-validation across the TLINKs, ignoring which documents the TLINKs came from. The (Mani et al., 2006) approach compared two different learning strategies: (i) learning from the TLINKs found in the annotated documents (the so-called "unclosed" approach, using ME), and comparing against the human annotations; and (ii) a "closed" approach (ME-C, for Maximum Entropy learning with closure) that first ran temporal reasoning using Sputlink (described below) on each document, and then learning from the resulting TLINKs, and comparing against closed human annotations. Notice that this "closed" classification task seems harder, since any TLINK, even TLINKs introduced by closure that relate elements (events or times) far apart in the document, has to be classified.

For temporal reasoning, we used a temporal closure component SputLink (Verhagen 2004),

[3]We show the counts in the vectors generated for TLINKs from the ATC, rather than the counts in the raw ATC itself. As a result of normalizations carried out by the vector-generation program, some TLINKs in the ATC are dropped.

[4]sourceforge.net/projects/carafe

that takes known temporal relations in a text and derives new implied relations from them, in effect making explicit what was implicit. SputLink was inspired by (Setzer and Gaizauskas 2000) and is based on Allen's interval algebra, taking into account the limitations on that algebra that were pointed out by (Vilain et al., 1989). It is basically a constraint propagation algorithm that uses a transitivity table to model the compositional behavior of all pairs of relations in a document. SputLink's transitivity table is represented by 745 axioms.

Relation	**Event-Event**	**Event-Time**
IBEFORE	110	9
BEGINS	142	102
ENDS	175	161
SIMULTANEOUS	1328	57
INCLUDES	788	2804 (65.99%)
BEFORE	2757 (52.01%)	1116
TOTAL	**5300**	**4249**

Table 1. TLINK Class Distributions in ATC Corpus (12,750 Events, 2,114 Times)

	UNCLOSED (ME)						**CLOSED (ME-C)**					
	Event-Event (5,300)			**Event-Time (4,249)**			**Event-Event (13,985)**			**Event-Time (7,664)**		
Accuracy:	61.79 (52.0)			84.21 (65.4)			76.56 (54.6)			83.23 (45.2)		
Relation	**Prec**	**Rec**	**F**	**Prec**	**Rec**	**F**	**Prec**	**Rec**	**F**	**Prec**	**Rec**	**F**
IBEFORE	50.83	28.97	36.91	0	0	0	60.01	39.33	47.52	91.00	39.12	54.73
BEGINS	58.70	40.53	47.95	32.86	29.29	30.97	81.83	61.81	70.43	58.48	43.61	49.96
ENDS	70.04	64.54	67.17	62.87	57.83	60.24	81.90	68.27	74.46	65.20	51.47	57.53
SIMULTANEOUS	52.32	55.60	53.91	27.15	20.07	23.08	58.81	54.45	56.55	39.56	32.08	35.43
INCLUDES	46.89	41.62	44.10	87.33	89.71	88.51	75.54	77.10	76.31	86.57	87.08	86.83
BEFORE	70.17	72.72	71.42	73.00	70.35	71.65	82.02	85.71	83.82	82.50	86.06	84.24

Table 2. Machine Learning Evaluated by Instance-level Partitioning

	UNCLOSED (ME)						**CLOSED (ME-C)**					
	Event-Event (5,300)			**Event-Time (4,249)**			**Event-Event (13,985)**			**Event-Time (7,664)**		
Accuracy:	59.68 (51.7)			82.47 (65.5)			51.14 (54.1)			71.99 (51.3)		
Relation	**Prec**	**Rec**	**F**	**Prec**	**Rec**	**F**	**Prec**	**Rec**	**F**	**Prec**	**Rec**	**F**
IBEFORE	61.02	23.87	34.31	0	0	0	42.27	12.00	18.70	0	0	0
BEGINS	51.28	34.41	41.19	43.35	26.23	32.68	50.23	15.43	23.61	33.66	14.82	20.58
ENDS	72.27	52.04	60.51	55.26	51.11	53.11	49.16	38.54	43.21	35.67	26.96	30.71
SIMULTANEOUS	49.76	51.15	50.45	47.04	38.70	42.46	32.15	45.23	37.59	27.40	18.28	21.93
INCLUDES	43.47	28.36	34.32	87.77	90.10	88.92	32.02	16.13	21.46	73.72	83.15	78.15
BEFORE	67.09	76.80	71.62	76.30	77.75	77.02	63.90	70.23	66.91	74.29	68.94	71.51

Table 3. Machine Learning Evaluated by Document-level Partitioning

LINK	**GOLD**	**CLASSIFIER**	**CONFIDENCE**	**CLOSURE**
placement_e5 raise_e4	SIMULTANEOUS	BEFORE	0.726	(BEFORE)
said_e7 said_e1	SIMULTANEOUS	**SIMULTANEOUS**	0.594	
said_e8 said_e1	SIMULTANEOUS	**SIMULTANEOUS**	0.594	
has_e29 option_e11	SIMULTANEOUS	**SIMULTANEOUS**	0.736	(**SIMULTANEOUS**)
redeem_e13 conversion_e14	BEFORE	**BEFORE**	0.979	**BEFORE**
conversion_e14 takes_e16	SIMULTANEOUS	BEFORE	0.930	(BEFORE)
said_e17 said_e1	SIMULTANEOUS	**SIMULTANEOUS**	0.594	
fixed_e18 set_e20	SIMULTANEOUS	BEFORE	0.968	BEFORE

Table 4. Illustration of Greedy Method

5.2 Replicating Approach I

Re-running such an evaluation, using a more up-to-date set of the 73 AQUAINT documents, produces the results shown in Table 2. In our learning experiments here, we use five-fold cross-validation (and the scores shown are averages across folds). The number of vectors for event-event links goes from 5,300 before closure to 13,985 after closure, while the number of

event-time links goes from 4,249 to 7,664. Table 2 shows that the ME-C approach significantly outperforms ME for event-event links.

Each of these results was significantly better than the majority class (shown in parentheses), and ME outperformed a variety of other classifiers, including the SMO support-vector machine and the naive Bayes tools in WEKA[5]. SMO performance (but not naive Bayes) was comparable with ME, with SMO trailing it in a few cases.

With the event-time data, the ratios before closure of the two most frequent classes are 65.99% for INCLUDES and 26.26% for BEFORE. After closure, the ratios are 51.06% for INCLUDES and 40.86% for BEFORE. The more balanced ratios after closure make the problem harder. With the event-event data, the ratios of the three most frequent classes before closure are 52.01% for BEFORE, 25.05% for SIMULTANEOUS, and 14.86% for INCLUDES, but after closure, while we still have a similar proportion of BEFORE (54.58%), we now have 21.23% for INCLUDES and 17.65% for SIMULTANEOUS.

5.3 Duplicate Vectors

Note that the number of closed vectors as well as the accuracy figures reported in (Mani et al., 2006) are substantially higher, with ME-C accuracy of 93.1% for event-event links and 88.25% for event-time links.

Apart from the differences in the data set, the discrepancy is due to the presence of a large number of duplicate vectors in their data, arising mainly as a result of collapsing inverse links after closure. Filtering out the duplicates, along with improvements to the vector generation program, results in far fewer closed vectors, less skew, and a lower accuracy.

6 Approach II: Partitioning by Documents

A problem with the (Mani et al., 2006) evaluation method is that it ignored which documents the vectors came from. Instead of evaluating on instances without regard to the document boundaries, we report here on a different method. Here the training and test documents, as opposed to training and test TLINK instances as earlier, were both partitioned by fold (into five-folds). The number of training (likewise, test) documents was approximately equal across folds. These new results are shown in Table 3.

While accuracy for event-event links was 61.79% at the instance level for unclosed data, it now drops slightly to 59.68%, with a similar 2-point drop in accuracy in event-time linking. Much more striking, however, is the drop in the results from closure. Here, closure no longer outperforms the majority class on event-event links, and for both event-event and event-time links, accuracy is worse than unclosed.

The poorer performance of ME-C in document-level compared to instance-level partitioning can be explained as follows. In instance-level partitioning, even when the event-event (or event-time) pairs in training and test vectors are distinct, there can be shared context across those vectors when they originate from the same document. For example, given a chain of four events A, B, C, and D linked by BEFORE, if B-C (labeled by the human) is in training and A-D (inferred by closure) is in test, there could still be overlapping features due to, say, B-C and A-D having the same (past) tense. Such shared context is absent when the testing is on documents that are distinct from training documents.

This shared context can explain the substantial improvement in ME-C over ME in instance-level partitioning; however, more analysis of the effect of closure on instances is clearly needed. In the case of event-time links, any advantages in closure accruing from shared context are likely to be offset by the aforementioned increased hardness of the problem.

7 Approach III: Global Inference

A fundamental problem underlying the previous approaches is that the classifier does not take into account any dependencies between TLINKs. For example, a classifier may label the TLINK <X,Y> as BEFORE (where X and Y are events or times). Given the pair <X,Z>, such a classifier has no idea if <Y,Z> has been classified as BEFORE, in which case, through closure, <X,Z> should be classified as BEFORE. This can result in the classifier producing an inconsistently annotated document.

To address this problem, we propose a greedy method for ensuring global consistency. At training time a simple classifier is learned based on pair-wise temporal relations in the training data. At test time, the test instances are ranked by confidence and the temporal closure axioms are

[5] sourceforge.net/projects/weka/

applied in an iterative fashion starting with the most confident instances.

Our approach relies on temporal closure to validate the TLINKs generated by the classifier (ME). All those TLINKs are put on a queue Q ordered by confidence score (where zero is the least confident and 1 is the most confident). The other data structure used is a set L of result links which is initially empty. The procedure consists of a loop in which (i) the link k with the highest score at or above a threshold is popped off Q and added to L, (ii) k is validated with respect to the other links in L, and (iii) k is retracted from S if validation fails. The procedure stops when there are no links at or above the confidence threshold.

Validation of k consists of two steps: comparing k to links already in L and running temporal closure. As a result, L will at any state be consistent and contain all information that can be inferred by using the closure axioms.

To illustrate, here is an example document (unclosed wsj_0106) processed using ME-G (Maximum Entropy classifier with Greedy Method) with the default confidence threshold of 0.95.

```
<DOCNO> WSJ891102-0086 </DOCNO>
<DD>11/02/89</DD>
<TEXT>
ROGERS COMMUNICATIONS Inc. said_e1 it plans_e2 to raise_e4 175 million to 180 million Canadian dollars (US $148.9 million to $153.3 million) through a private placement_e5 of perpetual preferred shares. Perpetual preferred shares aren't retractable by the holders, the company said_e7. Rogers said_e8 the shares will be convertible_e27 into Class B shares, but that the company has_e29 the option_e11 to redeem_e13 the shares before a conversion_e14 takes_e16 place. A spokesman for the Toronto cable television and telecommunications concern said_e17 the coupon rate hasn't yet been fixed_e18, but will probably be set_e20 at around 8%. He declined_e21 to discuss_e23 other terms of the issue.
</TEXT>
```

Table 4 shows the link, the gold standard label from ATC, the classifier label and its confidence, and the label resulting from the greedy closure at the confidence threshold of 0.95. We also show the impact of dropping the confidence threshold sharply to 0.6, the additional labels added in the latter case being shown in parentheses.

8 Conclusion

At the time of writing, we have finished an implementation of Approach III, but have not yet carried out a satisfactory evaluation of it. Note that for use of any of the these approaches in practice, it is not enough to solve Task B (the labeling task, the focus of this paper), but also Task A (the linking task). We are focused on both these evaluation challenges.

References

James Allen. 1984. Towards a General Theory of Action and Time. Artificial Intelligence, 23, 2, 123-154.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pusetjovsky,. 2006. Machine Learning of Temporal Relations. Proceedings of ACL'2006.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, David Day, Lisa Ferro, Robert Gaizauskas, Marcia Lazo, Andrea Setzer, and Beth Sundheim. 2003. The TimeBank Corpus. Corpus Linguistics, 647-656.

James Pustejovsky, Bob Ingria, Roser Sauri, Jose Castano, Jessica Littman, Rob Gaizauskas, Andrea Setzer, G. Katz, and I. Mani. 2005. The Specification Language TimeML. In I. Mani, J. Pustejovsky, and R. Gaizauskas, (eds.), The Language of Time: A Reader. Oxford University Press.

Andrea Setzer and Robert Gaizauskas. 2000. Annotating Events and Temporal Information in Newswire Texts. Proceedings of LREC-2000, 1287-1294.

Marc Verhagen. 2004. Times Between The Lines. Ph.D. Dissertation, Department of Computer Science, Brandeis University.

Marc Vilain, Henry Kautz, and Peter Van Beek. 1989. Constraint propagation algorithms for temporal reasoning: A revised report. In D. S. Weld and J. de Kleer (eds.), Readings in Qualitative Reasoning about Physical Systems, Morgan-Kaufman, 373-381.