LAMP-TR-138
CS-TR-4844
UMIACS-TR-2006-58

DECEMBER 2006

SUMMARIZATION-INSPIRED TEMPORAL-RELATION EXTRACTION: TENSE-PAIR TEMPLATES AND TREEBANK-3 ANALYSIS

Bonnie Dorr
Department of Computer Science
University of Maryland
College Park
MD 20742-3275
bonnie@umiacs.umd.edu

Terry Gaasterland
Scripps Institution of Oceanography
University of California
San Diego, La Jolla, CA, 92093
gaasterland@ucsd.edu

Abstract

This document describes the information used for summarization-inspired temporal-relation extraction [Dorr and Gaasterland, 2007]. We present a set of tense/aspect extraction templates that are applied to a Penn Treebank-style analysis of the input sentence. We also present an analysis of tense-pair combinations for different temporal connectives based on a corpus analysis of complex tense structures in Treebank-3. Finally, we include analysis charts and temporal relation tables for all combinations of intervals/points for each legal BTS combinations.

This work has been supported, in part, under the GALE program of the Defense Advanced Research Projects Agency, Contract Number HR0011-06-2-0001. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **DEC 2006**	2. REPORT TYPE	3. DATES COVERED **00-12-2006 to 00-12-2006**

4. TITLE AND SUBTITLE
Summarization-Inspired Temporal-Relation Extraction: Tense-Pair Templates and Treebank-3 Analysis

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Maryland,Department of Computer Science,College Park,MD,20742-3275

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**11**	

```
(S1
  (S (NP-SBJ (NP (NNP John)))
     (VP (VBD caught) (NP (PRP his) (NN plane))
       (SBAR-TMP (IN before)
         (S (NP-SBJ (NP (NNP Mary))) (VP (VBD arrived)))))))
```

Figure 1: Penn Treebank Representation of Temporally Related Matrix/Adjunct Pair

1 Introduction

This document describes the information used for summarization-inspired temporal-relation extraction [Dorr and Gaasterland, 2007].

First, we present a set of tense/aspect extraction templates that are applied to a Penn Treebank-style analysis of the input sentence (e.g., the output of parsers by [Collins, 1996] or [Charniak, 2000]). For example, a Penn Treebank-style analysis of *John caught his plane before Mary arrived* is shown in Figure 1, where the matrix clause starts with the first S and the adjunct clause begins with the SBAR-TMP node. Note that we examine only the matrix/adjunct pairs that contain an SBAR-TMP node, in an attempt to separate out temporally related clauses from those that are causally related.

Next, we present an analysis of tense-pair combinations for different temporal connectives based on a corpus analysis of complex tense structures in Treebank-3.

Finally, we include analysis charts for all combinations of intervals/points for each legal BTS combinations. Each analysis chart is associated with a more succinct temporal relation table.

2 Templates for Computing BTS from Treebank-Style Parse Trees

The Lisp variable *ts-table* provides the templates necessary for computing *tense structures* for computing the BTSs. Wildcards are indicated with question marks (?). The software that uses these templates is available at: `http://www.umiacs.umd.edu/~bonnie/CDTS-Solution`

```
(defparameter *ts-table*
  '(("UNCLASSIFIED"                     ; "Unclassified"
    ())
    ("PRESENT PROG"                     ; "Present Progressive: S,R,E"
     ((VBP AM   ) (VBG ?))
     ((VBP |'M| ) (VBG ?))
     ((VBP ARE  ) (VBG ?))
     ((VBP |'RE|) (VBG ?))
     ((VBZ IS   ) (VBG ?))
     ((VBZ |'S| ) (VBG ?)))
    ("PRES PERF"                        ; "Present Perfect: E_S,R"
     ((VBZ HAS  ) (VBN ?))
     ((VBZ |'S| ) (VBN ?))
     ((VBP HAVE ) (VBN ?))
     ((VBP |'VE|) (VBN ?)))
    ("PRESENT SIMPLE STATE"             ; "Present Simple State: S,R,E"
     ((VBP AM   ))
     ((VBP |'M| ))
     ((VBP ARE  ))
     ((VBP |'RE|))
     ((VBZ IS   ))
```

```
 ((VBZ |'S| ))
 ((MD CAN   ) (VB BE))
 ((MD CA   ) (VB BE))
 ((MD MAY   ) (VB BE))
 ((MD MUST  ) (VB BE))
 ((MD NEED  ) (VB BE))
 ((MD MIGHT ) (VB BE))
 ((MD OUGHT ) (VB BE))
 ((MD COULD ) (VB BE))
 ((MD SHOULD) (VB BE))
 ((MD WOULD ) (VB BE))
 ((MD |'D|  ) (VB BE)))
("PRESENT SIMPLE"                   ; "Present Simple: S,R,E"
 ((VBP ?    ))
 ((VBZ ?    ))
 ((MD CAN   ) (VB ?))
 ((MD CA   )  (VB ?))
 ((MD DO   )  (VB ?))
 ((MD DOES  )  (VB ?))
 ((MD MAY   ) (VB ?))
 ((MD MUST  ) (VB ?))
 ((MD NEED  ) (VB ?))
 ((MD MIGHT ) (VB ?))
 ((MD OUGHT ) (VB ?))
 ((MD COULD ) (VB ?))
 ((MD SHOULD) (VB ?))
 ((MD WOULD ) (VB ?))
 ((MD |'D|  ) (VB ?)))
;;"Past"
("PAST PROG"                        ; "Past Progressive: E,R_S"
 ((VBD WAS  ) (VBG ?))
 ((VBD WERE ) (VBG ?)))
("PAST PERF"                        ; "Past Perfect: E_R_S"
 ((VBD HAD  ) (VBN ?))
 ((VBD |'D| ) (VBN ?)))
("PAST SIMPLE STATE"                ; "Past Simple State: E,R_S"
 ((VBD WAS  ))
 ((VBD WERE ))
 ((MD COULD ) (VB BE))
 ((MD MAY   ) (VB HAVE) (VBN BEEN))
 ((MD MUST  ) (VB HAVE) (VBN BEEN))
 ((MD NEED  ) (VB HAVE) (VBN BEEN))
 ((MD MIGHT ) (VB HAVE) (VBN BEEN))
 ((MD COULD ) (VB HAVE) (VBN BEEN))
 ((MD SHOULD) (VB HAVE) (VBN BEEN))
 ((MD WOULD ) (VB HAVE) (VBN BEEN))
 ((MD |'D|  ) (VB HAVE) (VBN BEEN)))
("PAST SIMPLE"                      ; "Past Simple: E,R_S"
 ((VBD ?    ))
 ((MD DID   ) (VB ?))
 ((MD COULD ) (VB HAVE) (VBN ?))
 ((MD MAY   ) (VB HAVE) (VBN ?))
 ((MD MUST  ) (VB HAVE) (VBN ?))
 ((MD NEED  ) (VB HAVE) (VBN ?))
 ((MD MIGHT ) (VB HAVE) (VBN ?))
 ((MD COULD ) (VB HAVE) (VBN ?))
 ((MD SHOULD) (VB HAVE) (VBN ?))
 ((MD WOULD ) (VB HAVE) (VBN ?))
 ((MD |'D|  ) (VB HAVE) (VBN ?)))
;;"Future"
("FUT PROG"                         ; "Future Progressive: S_R,E"
 ((MD WILL  ) (VB BE  ) (VBG ?))
 ((MD |'LL| ) (VB BE  ) (VBG ?))
 ((MD WO    ) (VB BE  ) (VBG ?))
 ((MD SHALL ) (VB BE  ) (VBG ?)))
("FUT PERF"                         ; "Future Perfect: S_E_R"
 ((MD WILL  ) (VB HAVE) (VBN ?))
 ((MD |'LL| ) (VB HAVE) (VBN ?))
```

```
  ((MD WO    ) (VB HAVE) (VBN ?))
  ((MD SHALL ) (VB HAVE) (VBN ?)))
 ("FUT SIMPLE STATE"                ; "Future Simple State: S_R,E"
  ((MD WILL  ) (VB BE  ))
  ((MD |'LL| ) (VB BE  ))
  ((MD WO    ) (VB BE  ))
  ((MD SHALL ) (VB BE  )))
 ("FUT SIMPLE"                      ; "Future Simple: S_R,E"
  ((MD WILL  ) (VB ?   ))
  ((MD |'LL| ) (VB ?   ))
  ((MD SHALL ) (VB ?   ))
  ((MD WO    ) (VB ?   )))
 ;; Gerund
 ("SIMPLE GERUND"                   ; "Simple Gerund: R,E"
  ((VBG ?    )))
 ))
```

3 Analysis of tense-pair Combinations for Different Temporal Connectives in TreeBank-3

This section contains an analysis of tense-pair combinations for different temporal connectives based on a corpus analysis of complex tense structures in Treebank-3.

3.1 Tense Pairs with Respect to Temporal Connective AFTER

Total	**Matrix/Adjunct Pairs**
151	[past simp]/[past simp]
35	[past simp state]/[past simp]
17	[past simp]/[past simp state]
10	[past simp state]/[past simp state]
7	[simp gerund]/[past simp], [past simp]/[past perf]
6	[past perf]/[past simp]
5	[fut simp]/[pres simp], [pres simp]/[pres simp state]
4	[pres simp state]/[pres simp state]
3	[fut simp state]/[pres simp state], [fut simp state]/[pres simp], [fut simp]/[pres simp state], [pres simp]/[pres perf], [pres simp]/[pres simp], [past perf]/[past simp state], [past simp state]/[past perf]
1	[past simp]/[simp gerund], [fut simp state]/[pres perf], [fut simp]/[pres perf], [simp gerund]/[pres perf], [simp gerund]/[pres simp state], [simp gerund]/[past perf], [simp gerund]/[simp gerund], [pres simp state]/[pres simp], [past prog]/[past perf], [past prog]/[past simp]

3.2 Tense Pairs with Respect to Temporal Connective AS

Total	Matrix/Adjunct Pairs for AS
173	[past simp]/[past simp]
58	[pres simp]/[pres simp]
21	[past simp state]/[past simp]
18	[pres simp state]/[pres simp]
15	[pres prog]/[pres simp]
14	[fut simp]/[pres simp]
10	[pres perf]/[pres simp]
9	[simp gerund]/[pres simp], [simp gerund]/[past simp]
8	[past simp]/[past prog]
6	[pres perf]/[pres perf], [pres simp]/[pres prog], [past simp]/[past simp state]
4	[pres simp state]/[pres simp state], [past prog]/[past simp], [past perf]/[past simp]
3	[fut simp state]/[pres simp]
2	[pres simp]/[pres simp state], [past simp state]/[past prog]
1	[fut prog]/[pres simp], [pres prog]/[pres prog], [fut simp]/[pres simp state], [pres simp state]/[pres prog], [pres simp state]/[pres perf], [pres simp]/[pres perf], [past prog]/[past prog], [past prog]/[past simp state], [past simp state]/[past simp state], [past simp]/[past perf]

3.3 Tense Pairs with Respect to Temporal Connective BEFORE

Total	Matrix/Adjunct Pairs for BEFORE
34	[past simp]/[past simp]
22	[pres simp]/[pres simp]
13	[past simp state]/[past simp]
10	[fut simp]/[pres simp]
9	[pres simp state]/[pres simp]
7	[pres simp]/[pres simp state], [past simp]/[past simp state]
6	[past prog]/[past simp]
5	[past perf]/[past simp], [past simp state]/[past simp state]
4	[pres prog]/[pres simp]
3	[fut simp state]/[pres simp], [pres simp state]/[pres simp state]
2	[fut simp state]/[pres simp state], [fut simp state]/[fut simp], [fut simp]/[pres simp state]
1	[simp gerund]/[pres simp], [simp gerund]/[past simp], [simp gerund]/[simp gerund], [pres perf]/[pres perf], [pres simp]/[pres perf], [past perf]/[past simp state], [past simp state]/[past perf], [past simp]/[past perf]

3.4 Tense Pairs with Respect to Temporal Connective ONCE

Total	Matrix/Adjunct Pairs for ONCE
18	[pres simp]/[pres simp]
13	[pres simp]/[pres simp state]
6	[fut simp]/[pres simp], [past simp]/[past simp]
5	[fut simp]/[pres simp state]
3	[pres simp state]/[pres simp], [past simp state]/[past simp]
2	[fut simp state]/[pres simp], [pres simp state]/[pres simp state]
1	[fut perf]/[pres simp state], [fut simp state]/[pres simp state], [simp gerund]/[pres perf], [pres simp state]/[pres perf], [past simp state]/[past simp state], [past simp]/[past simp state]

3.5 Tense Pairs with Respect to Temporal Connective SINCE

Total	Matrix/Adjunct Pairs for SINCE
5	[past simp state]/[past simp]
2	[pres perf]/[pres perf], [past simp]/[past simp state], [past simp]/[past simp]
1	[simp gerund]/[past simp], [pres simp state]/[pres perf], [pres simp state]/[pres simp state], [pres simp]/[pres simp state], [pres simp]/[pres simp], [past perf]/[past simp]

3.6 Tense Pairs with Respect to Temporal Connective UNTIL

Total	**Matrix/Adjunct Pairs for UNTIL**
16	[past simp]/[past simp]
9	[fut simp]/[pres simp state], [pres simp]/[pres simp]
7	[fut simp]/[pres simp]
6	[fut simp state]/[pres simp], [past simp state]/[past simp]
5	[pres simp]/[pres simp state]
3	[fut simp state]/[pres simp state], [fut simp]/[pres perf], [simp gerund]/[pres simp state], [simp gerund]/[pres simp], [pres simp state]/[pres simp state], [pres simp state]/[pres simp], [pres simp]/[pres perf], [past simp]/[past perf]
2	[pres prog]/[pres simp state], [past prog]/[past simp state], [past simp]/[past simp state]
1	[pres simp state]/[pres perf], [past prog]/[past simp], [past perf]/[past simp state], [past perf]/[past simp]

3.7 Tense Pairs with Respect to Temporal Connective WHEN

Total	**Matrix/Adjunct Pairs for WHEN**
254	[past simp]/[past simp]
116	[pres simp]/[pres simp]
76	[past simp state]/[past simp]
52	[past simp]/[past simp state]
54	[pres simp state]/[pres simp]
37	[pres simp]/[pres simp state]
21	[fut simp]/[pres simp]
20	[pres simp state]/[pres simp state]
19	[past simp state]/[past simp state]
14	[simp gerund]/[pres simp]
12	[pres simp]/[pres prog]
11	[pres simp]/[simp gerund]
8	[past perf]/[past simp], [past simp]/[past prog]
7	[pres simp state]/[pres prog], [past prog]/[past simp]
6	[pres prog]/[pres simp], [pres perf]/[pres simp]
5	[fut simp]/[pres simp state], [simp gerund]/[past simp], [pres simp]/[pres perf]
4	[fut simp state]/[pres simp], [simp gerund]/[pres simp state], [past perf]/[past simp state]
3	[simp gerund]/[past simp state], [past prog]/[past simp state]
2	[fut simp state]/[pres simp state], [fut simp]/[pres perf], [pres prog]/[pres simp state], [past simp state]/[past prog]
1	[past simp]/[simp gerund], [fut prog]/[pres simp], [pres prog]/[pres prog], [pres prog]/[pres perf], [simp gerund]/[pres prog], [simp gerund]/[pres perf], [simp gerund]/[simp gerund], [pres perf]/[pres perf], [pres perf]/[pres simp state], [pres simp state]/[simp gerund], [past perf]/[past prog], [past simp]/[past perf]

3.8 Tense Pairs with Respect to Temporal Connective WHILE

Total	**Matrix/Adjunct Pairs for WHILE**
27	[past simp]/[past simp]
16	[pres simp]/[simp gerund]
13	[past simp]/[simp gerund]
12	[past simp]/[past simp state]
11	[pres simp]/[pres simp]
7	[simp gerund]/[simp gerund]
5	[past simp]/[past prog]
4	[simp gerund]/[past simp], [pres perf]/[pres perf]
3	[simp gerund]/[pres simp], [pres perf]/[simp gerund], [pres simp]/[pres simp state], [past simp state]/[past simp]
2	[pres prog]/[pres simp], [pres prog]/[simp gerund], [pres perf]/[pres simp], [pres simp state]/[pres prog], [pres simp state]/[pres simp state], [pres simp state]/[simp gerund], [pres simp]/[pres prog], [past prog]/[past simp], [past prog]/[simp gerund], [past simp state]/[past prog], [past simp state]/[past simp state]
1	[fut simp state]/[pres simp], [fut simp]/[pres simp state], [fut simp]/[pres simp], [fut simp]/[simp gerund], [simp gerund]/[pres simp state], [simp gerund]/[past simp state], [pres perf]/[pres simp state], [pres simp state]/[pres perf], [pres simp]/[pres perf], [past perf]/[past simp state]

4 Analysis Chart for Past/Past BTS Combination

To build a full implementation of the method for extracting Allen's temporal relations, an analysis chart must be built for all combinations of intervals/points for each legal BTS combinations. The analysis charts contain combinations of verbs from different aspectual categories: state (*be angry/happy*), extended activity (*walk*), point activity (*wink*), accomplishment (*write a letter*), and achievement (*win the race*).

In addition, the progressive and simple forms are included for each verb. In producing the analysis chart, certain linguistic generalizations became apparent. In particular, we observed that the *activity/achievement/accomplishment* distinction did not affect the connecting word meanings. Thus, we were able to construct a more succinct temporal relation table for each analysis chart.

Below we include the analysis chart and temporal relation table for the past/past and future/present BTS Combination.

4.1 Analysis Chart and Temporal Relation Table for the Past/Past BTS Combination

This section includes an analysis chart and temporal relation table for the Past/Past BTS combination for *after*, *before*, and *while*, e.g., *She won the race while John wrote a letter*.

Matrix	**AFTER**	**BEFORE**	**WHILE**	**Adjunct**
Mary was winning the race *Ach, Prog:* ●—●	>	<	= o oi s d f	John was writing a letter *Acc, Prog:* ●—●
Mary was winning the race *Ach, Prog:* ●—●	oi mi f >	o m f <	″	John was winking *Act(pt), Prog:* ●—●, ●—○
John was winking *Act(pt), Prog:* ●—●, ●—○	>	<	″	Mary was winning the race *Ach, Prog:* ●—●
John was winking *Act(pt), Prog:* ●—●, ●—○	oi mi f >	o m fi <	″	Mary was walking *Act(ext), Prog:* ●—●, ●—○
Mary was winning the race *Ach, Prog:* ●—●	>	<	= o oi s d f	John wrote a letter *Acc, Simp:* ●
Mary was winning the race *Ach, Prog:* ●—●	″	″	″	John winked *Act(pt), Simp:* ●
John was winking *Act(pt), Prog:* ●—●, ●—○	″	″	″	Mary won the race *Ach, Simp:* ●
John was winking *Act(pt), Prog:* ●—●, ●—○	″	″	= oi s d f <	Mary walked *Act(ext), Simp:* ●—●
Mary was winning the race *Ach, Prog:* ●—●	oi mi f >	o m fi <	″	John was angry *State, Simp:* ●—●, ●—○
Mary was winking *Act(pt), Prog:* ●—●, ●—○	″	″	″	John was angry *State, Simp:* ●—●, ●—○
Mary won the race *Ach, Simp:* ●	>	<	= s d f fi	John was writing a letter *Acc, Prog:* ●—●
Mary won the race *Ach, Simp:* ●	oi mi f >	o m fi <	″	John was winking *Act(pt), Prog:* ●—●, ●—○
John winked *Act(pt), Simp:* ●	>	<	″	Mary was winning the race *Ach, Prog:* ●—●
John winked *Act(pt), Simp:* ●	oi mi f >	o m fi <	″	Mary was walking *Act(ext), Prog:* ●—●, ●—○
Mary won the race *Ach, Simp:* ●	>	<	= s d f fi	John wrote a letter *Acc, Simp:* ●
Mary won the race *Ach, Simp:* ●	″	″	″	John winked *Act(pt), Simp:* ●
John winked *Act(pt), Simp:* ●	″	″	″	Mary won the race *Ach, Simp:* ●
John winked *Act(pt), Simp:* ●	″	″	″	Mary walked *Act(ext), Simp:* ●—●
Mary won the race *Ach, Simp:* ●	oi mi f >	o m fi <	= s d f	John was angry *State, Simp:* ●—●, ●—○
Mary winked *Act(pt), Simp:* ●	″	″	″	John was angry *State, Simp:* ●—●, ●—○
John was angry *State, Simp:* ●—●, ●—○	>	<	= o oi s d f	Mary was winning the race *Ach, Prog:* ●—●
John was angry *State, Simp:* ●—●, ●—○	oi mi f >	o m fi <	″	Mary was walking *Act(ext), Prog:* ●—●, ●—○
John was angry *State, Simp:* ●—●, ●—○	>	<	″	Mary won the race *Ach, Simp:* ●
John was angry *State, Simp:* ●—●, ●—○	″	″	″	Mary walked *Act(ext), Simp:* ●—●
John was angry *State, Simp:* ●—●, ●—○	oi mi f >	o m fi <	″	Mary was happy *State, Simp:* ●—●, ●—○

The corresponding temporal relation table for the Past/Past tense combination is given here, where closed intervals are referred to as **C**, open intervals are referred to as **O**, and point-intervals are referred to as **P**.

Matrix/Adjunct	**AFTER**	**BEFORE**	**WHILE**
C/C	>	<	= o oi s d f
C/O	oi mi f >	o m fi <	= o oi s d f
C/P	>	<	= o oi s d f
O/C	>	<	= o oi s d f
O/O	oi mi f >	o m fi <	= o oi s d f
O/P	>	<	= o oi s d f
P/C	>	<	= s d f
P/O	oi m f >	o m fi <	= s d f
P/P	>	<	= s d f fi

4.2 Analysis Chart and Temporal Relation Table for the Future/Present BTS Combination

This section includes an analysis chart and temporal relation table for the Future/Present BTS combination for *after*, *before*, and *while*, e.g., *She will win the race while John writes a letter*.

Matrix	AFTER	BEFORE	WHILE	Adjunct
Mary will be winning the race *Ach, Prog:* •—•	oi f >	o fi <	= o s d f	John is writing a letter *Acc, Prog:* •—•
Mary will be winning the race *Ach, Prog:* •—•	″	″	″	John is winking *Act(pt), Prog:* •—•, •—○
John will be winking *Act(pt), Prog:* •—•, •—○	″	″	″	Mary is winning the race *Ach, Prog:* •—•
John will be winking *Act(pt), Prog:* •—•, •—○	″	″	″	Mary is walking *Act(ext), Prog:* •—•, •—○
Mary will be winning the race *Ach, Prog:* •—•	>	<	= oi s d f	John writes a letter *Acc, Simp:* •
Mary will be winning the race *Ach, Prog:* •—•	″	″	″	John winks *Act(pt), Simp:* •
John will be winking *Act(pt), Prog:* •—•, •—○	″	″	″	Mary wins the race *Ach, Simp:* •
John will be winking *Act(pt), Prog:* •—•, •—○	oi f >	o fi <	= o s d f	Mary walks *Act(ext), Simp:* •—•
Mary will be winning the race *Ach, Prog:* •—•	″	″	″	John is angry *State, Simp:* •—•, •—○
Mary will be winking *Act(pt), Prog:* •—•, •—○	″	″	″	John is angry *State, Simp:* •—•, •—○
Mary will win the race *Ach, Simp:* •	oi f >	<	= o oi s si d	John is writing a letter *Acc, Prog:* •—•
Mary will win the race *Ach, Simp:* •	″	″	″	John is winking *Act(pt), Prog:* •—•, •—○
John will wink *Act(pt), Simp:* •	″	″	″	Mary is winning the race *Ach, Prog:* •—•
John will wink *Act(pt), Simp:* •	″	″	″	Mary is walking *Act(ext), Prog:* •—•, •—○
Mary will win the race *Ach, Simp:* •	>	<	= o oi s si d	John writes a letter *Acc, Simp:* •
Mary will win the race *Ach, Simp:* •	″	″	″	John winks *Act(pt), Simp:* •
John will wink *Act(pt), Simp:* •	″	″	″	Mary wins the race *Ach, Simp:* •
John will wink *Act(pt), Simp:* •	oi f >	<	= o oi s si d	Mary walks *Act(ext), Simp:* •—•
Mary will win the race *Ach, Simp:* •	″	″	″	John is angry *State, Simp:* •—•, •—○
Mary will wink *Act(pt), Simp:* •	″	″	″	John is angry *State, Simp:* •—•, •—○
John will be angry *State, Simp:* •—•, •—○	oi f >	o fi <	= o s d f	Mary is winning the race *Ach, Prog:* •—•
John will be angry *State, Simp:* •—•, •—○	″	″	″	Mary is walking *Act(ext), Prog:* •—•, •—○
John will be angry *State, Simp:* •—•, •—○	″	″	= oi s d f	Mary wins the race *Ach, Simp:* •
John will be angry *State, Simp:* •—•, •—○	″	″	= o s d f	Mary walks *Act(ext), Simp:* •—•
John will be angry *State, Simp:* •—•, •—○	″	″	″	Mary is happy *State, Simp:* •—•

The corresponding temporal relation table for the Future/Present tense combination is given here, where closed intervals are referred to as **C**, open intervals are referred to as **O**, and point-intervals are referred to as **P**.

Matrix/Adjunct	AFTER	BEFORE	WHILE
C/C	f oi >	o fi <	= o s d f
C/O	f oi >	o fi <	= o s d f
C/P	>	<	= oi s d f
O/C	f oi >	o fi <	= o s d f
O/O	f oi >	o fi <	= o s d f
O/P	>	<	= oi s d f
P/C	f oi >	<	= o oi s si d
P/O	f oi >	<	= o oi s si d
P/P	>	<	= o oi s si d

References

[Charniak, 2000] Charniak, E., 2000. A maximum-entropy inspired parser. In: Proceedings of the First Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2000). Seattle, WA.

[Collins, 1996] Collins, M. J., 1996. A new statistical parser based on bigram lexical dependencies. In: Joshi, A., Palmer, M. (Eds.), Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics. Morgan Kaufmann Publishers, San Francisco, pp. 184–191.

[Dorr and Gaasterland, 2007] Dorr, B. J., Gaasterland, T., 2007. Exploiting Aspectual Features and Connecting Words for Summarization-Inspired Temporal-Relation Extraction. Journal of Information Processing and Management.