Andreas K. Orphanides. Extraction of Natural-Language Dates and Comparison of Dates in Hypothesis and Text to Identify Negative Textual Entailment. A Master's Paper for the M.S. in L.S degree. April, 2008. 66 pages. Advisor: Catherine Blake

We present a system to determine entailment, when a sentence (the text) implies a second sentence (the hypothesis). While some systems use temporal information to decide entailment, no study has measured the effectiveness of temporal features alone in entailment resolution. The system identifies natural-language dates, and precludes entailment solely by comparing dates in the hypothesis and text. Evaluation of date detection on three 800-pair corpora from the Recognising Textual Entailment (RTE) Challenges provided precision of 98.0% (RTE3devmt, 338/345), 96.6% (RTE3test, 198/205), and 99.7% (RTE1test, 336/337), and recall of 97.7% (RTE3devmt, 338/346), 99.0% (RTE3test, 198/200), and 99.1% (RTE1devmt 336/339). For sentence pairs with years, the proposed method improved entailment accuracy from 33 to 42/72 (RTE3devmt) and from 42 to 44/63 (RTE3test,) which corresponds to an overall improvement of 1.1% (RTE3devmt) and 0.1% (RTE3test). Our analysis suggests that matching temporal information with an event would further increase the entailment accuracy.

Headings:

Text mining -- Textual entailment

Text mining -- Temporal information

Natural language processing

EXTRACTION OF NATURAL-LANGUAGE DATES AND COMPARISON OF DATES IN HYPOTHESIS AND TEXT TO IDENTIFY NEGATIVE TEXTUAL ENTAILMENT

by
Andreas K. Orphanides

A Master's paper submitted to the faculty
of the School of Information and Library Science
of the University of North Carolina at Chapel Hill
in partial fulfillment of the requirements
for the degree of Master of Science in
Library Science.

Chapel Hill, North Carolina

April 2008

Approved by

_____

Catherine Blake

Table of Contents

Table of Figures

Table of Tables

1 Introduction

Dagan et al. (2005) define textual entailment as "a directional relationship between pairs of text expressions, denoted by T - the entailing 'Text', and H - the entailed 'Hypothesis'… [where] T entails H if the meaning of H can be inferred from the meaning of T, as would typically be interpreted by people." Consider this pair of texts from the third PASCAL Recognising Textual Entailment Challenge (Giampiccolo et al. 2007):

```
RTE3 Development Set pair 17:
T: Allen was renowned for his skill at scratch-building and
      creating scenery, and he pioneered the technique of
      weathering his models to make them look old and more
      realistic.
H: Allen introduced a new technique of creating realistic
      scenery.
```

Since T implies H, T entails H (stated differently, entailment between this pair holds, or the pair has positive entailment). On the other hand, consider this pair, from the same Challenge:

```
RTE3 Development Set pair 625:
T: Following a steady flow of leaks and statements on apparently
      incriminating discoveries, her office has been gagged by
      senior legal authorities.
H: Because of leaks, a gag was placed on Holland's office.
```

In this case, T does not entail H (or entailment does not hold, or the pair is said to have negative entailment), since information exists in the hypothesis (the office holder's name) that is not in the text. Entailment may fail to hold for other reasons; for instance, details from disparate ideas in the text may be conflated in the hypothesis:

```
RTE3 Development Set pair 25
T: Bountiful arrived after war's end, sailing into San Francisco
      Bay 21 August 1945. Bountiful was then assigned as hospital
      ship at Yokosuka, Japan, departing San Francisco 1 November
      1945.
H: Bountiful reached San Francisco on 1 November 1945.
```

Here, entailment does not hold because in the hypothesis, the idea of Bountiful reaching Japan is incorrectly paired with the date of Bountiful leaving Japan, resulting in a hypothesis that does not follow from the text.

Automated textual entailment recognition has many potential applications. Identifying an entailment relationship between two texts can establish a link between two ways of expressing an idea (Dagan et al., 2005), and recognizing an entailment relationship can suggest semantic associations between texts. The identification of a nonexistent or incomplete entailment relationship between texts can suggest that the meaning in the new text does not duplicate existing text (Monz & Rijke, 2001).

The problem of textual entailment relates to semantic entailment and inference in language. Condoravdi et al. introduced entailment as a distinct problem area i.e. "the detection of entailment...relations between texts") in 2003. Work on textual entailment began in earnest in 2004, with the establishment of the PASCAL Recognising Textual Entailment (RTE) Challenges (Dagan et al., 2005), and the development of the Microsoft Research Paraphrase Corpus (Dolan et al., 2005).

1.1 The PASCAL Recognising Textual Entailment Challenges

The PASCAL Recognising Textual Entailment Challenges were introduced in 2004 to motivate research in textual entailment. In each challenge, participants are provided with two sets of data, a development set and a test set. The sets each have 800 text/hypothesis

pairs, each pair consisting of one or more complete sentences in the text and in the hypothesis. The pairs were either taken from existing corpora or derived from freely available Web-based sources. The data sets consist of approximately a 50/50 split between pairs where entailment holds and those where it does not, as judged by three human annotators, who had to agree on a pair's entailment value for it to be included in the data set. Each pair is also labeled with a task type indicating a text application relevant to that pair, again as judged by the annotators. Each task type contains approximately a 50/50 split between entailed and non-entailed pairs. Participants know the task types and can exploit them in their systems (Dagan et al., 2005). Task types in the third Challenge were Information Extraction (IE), Information Retrieval (IR), Question Answering (QA), and Summarization (SUM) (Giampiccolo et al., 2007).

Challenge participants use the development data set (in which the entailment value of each pair is provided) during system development and tuning. Once they complete system development, they run their systems on the test data set (in which the participants do not know the entailment value of each pair); results are provided to the RTE judges, who measure system performance is measured in terms of accuracy, defined as the number of entailment predictions correct out of the number of pairs tested. Systems are optionally scored on a separate relative precision metric, which takes into account the system's confidence in each prediction. We consider only accuracy in our analysis of RTE systems; since the relative precision formula changed between RTE1 and RTE2, those scores cannot be compared (Dagan et al., 2005; Bar-Haim et al., 2006).

1.2 Previous approaches to the Recognising Textual Entailment Challenges

Approaches to recognizing textual entailment have evolved over the three PASCAL Recognising Textual Entailment Challenges. In RTE1, word overlap, lexical relations, and syntactic matching were each used on 13 of the 28 runs on the full corpus (46%). The accuracy of systems in RTE1 ranged from 49.5% to 58.6% on the whole corpus (Dagan et al., 2005). The lowest-scoring system (Perez & Alfonseca, 2005), however, achieved an accuracy score of 72% on RTE1's "Comparable Document" task type, using only word overlap features; this suggests that approaches unsuitable for the full corpus can achieve high accuracy on a well-defined subset thereof.

In RTE2, lexical relation identification was used in 32 out of 41 systems (78%) (Bar-Haim et al., 2006). Both the lowest scoring runs (Nicholson et al. (2006) with 50.88% and 52.88% accuracy scores), and the highest scoring (Tatu et al. (2006) with 73.75%; Hickl et al. (2006) with 75.38%) made use of this feature, and lexical relation is the only feature in common between Tatu et al. and Hickl et al., making this feature's contribution to performance unclear. In addition, Tatu et al. used three features, whereas Hickl used seven. This suggests that as features are added, each improves performance with diminishing effectiveness, and that carefully selecting features and providing for feature interaction can lead to strong performance.

In RTE3, the high scorers were Hickl & Bensley (2007) (80%), and Tatu & Moldovan (2007) (72.25%); the low scorers were Baral (49.63%) and Clark et al. (2007) (47.25%). Eight systems made use of anaphora resolution; all but one scoring above .6 accuracy (the exception scoring 59.75%). With the exception of Tatu's use of anaphora resolution,

the features used by Tatu and Clark are identical. The use of anaphora resolution alone does not account for the 25% difference in scores, especially in light of other systems that used more features than Tatu's (including anaphora resolution), and did not score as highly. This suggests that implementation of features, in addition to feature selection, affects system performance.

1.3 Temporal features in the Third Recognising Textual Entailment Challenge[1]

Although temporal feature analysis was not tracked as an approach in Giampiccolo et al. (2007), five systems in RTE3 (Bobrow et al.; Tatu & Moldovan; Rodrigo et al.; Adams et al.; MacCartney & Manning; all 2007) used temporal features. Tatu & Moldovan implemented full temporal tagging with the TARSQI (Verhagen et al., 2005) toolkit, using temporal features to establish logic representations. Rodrigo et al. observed that entailment holds between two temporal expressions if the hypothesis's temporal reference contains the text's reference (e.g., the date "1923" in the text entails the date range "the 1920s" in the hypothesis), which is corroborated by Delmonte et al. (2005). Adams et al. identified negative cases by using temporal relations in conjunction with grammatical dependency to compare event orders in the text and the hypothesis. That a temporal relationship is a necessary condition for a causal relationship (*Stanford Encyclopedia*, 2007) suggests that Adams et al. were correct to use temporal features for negation. MacCartney and Manning utilized temporal relations in their implementation of a natural logic.

[1] Although a number of systems in RTE1 (Delmonte et al., 2005; Raina et al., 2005) and RTE2 (Delmonte et al., 2006; Hickl et al., 2006; Tatu et al., 2006) made use of temporal features , we concentrate here on RTE3, where the number of systems making use of temporal features is higher.

1.4 Motivation beyond entailment

Temporal analysis is important for applications beyond entailment. Both Jang et al. (2004) and Wong et al. (2005) describe temporal analysis as a subset of information extraction (IE) and named entity recognition, and point out that temporal information is essential for question answering (QA) tasks, for questions that ask “when,” “how long,” or “in what sequence.” They also point out that temporal information can be used to order extracted text in summarization (SUM) tasks. These tasks represent three of the four task types in RTE3. Wong et al. identify temporal relations and derive temporal reasoning by combining analysis of temporal comparators such as “before” and “after” with identification of reference and reporting dates. Schilder (2004) extracts temporal information from noun and prepositional phrases. Roddick and Spilipoulou (2002) survey temporal knowledge discovery techniques in data mining applications, including temporal association discovery and temporal extensions to classification. These techniques use unstructured temporal information in text, including relative temporal keywords and natural-language dates. Jang et al. outline an automatic temporal tagging system for Korean, which uses a dictionary of temporal terms to identify and interpret relative, absolute, and duration-based temporal references. The TARSQI suite (Verhagen et al., 2005) consists of tools to automatically label temporal features according to the TIMEX3 temporal metadata standard, but its system requirements, which include Python, Perl, and SGML-tagged parts of speech, fall outside the scope of this project. We did use the temporal vocabulary from TARSQI’s GUTime tool as the basis for our own in a preliminary study; see section 2.1, page 11.

1.5 Research question and approach

Our goal is to investigate to what degree temporal information can improve performance in an entailment recognition system. We present a system that precludes entailment by comparing year-containing dates in the hypothesis to such dates in the text. We describe a date-extraction sequence that ensures that, for each date, all relevant information is extracted; and we provide a list of techniques for extracting different types of dates. We then provide a decision-making process that precludes entailment based on the dates identified in each entailment pair. We evaluate these methods on three data sets from the RTE challenges: the RTE3 development set (RTE3D), the RTE1 test set (RTE1T), and the RTE3 test set (RTE3T). We measure system performance on accuracy, number of predictions correct out of number made; and on recall, number of predictions correct in the subset containing temporal information extractable by the system. Finally, we propose techniques for matching actors (nouns) and actions (verbs) to the dates on which those actions were performed.

2 Methods

2.1 Preliminary investigations

We performed a pilot study to investigate the prevalence of temporal terms in entailment pairs. We created a temporal vocabulary extended from the vocabulary used by the TARSQI toolkit's GUTime tool (Verhagen et al., 2005). Our vocabulary included day names, month names, units of time such as "day," "month," and "year," temporal modifiers such as "before" and "after," and regular expression patterns that matched four-digit years (e.g. "1992") and decade statements (e.g., "1980s"). We used a SQL query to count sentence pairs in RTE3D that containing terms in the temporal vocabulary. The SQL query identified 493 pairs in RTE3D (61.6%) that contained at least one temporal term. Hand-counting revealed 456 true positives and 37 false positives, giving the query a precision (correctly identified pairs out of all pairs identified) of 92.5%. While all words in the temporal vocabulary were identified, the regular expression failed to match years in 4 pairs, for a recall (correctly identified pairs out of total pairs with temporal information) of (456/460) = 99.1%.

Our analysis revealed that 15.3% of RTE3D (122 pairs) contained temporal terms in the hypothesis. In 10 pairs, terms were incorrectly identified as temporal, for a precision of (112/122) = 91.8%. The regular expression failed to match a year in 1 hypothesis sentence, for a recall of (111/112) = 99.1%

We examined sentence pairs in RTE3D where a year in the hypothesis did not match years in the text. In 26 of the 27 such pairs (96.3%), entailment did not hold. This prompted the current approach of using temporal mismatches between the hypothesis and texts to identify cases of negative entailment.

Table 1: Pairs in RTE3 development set with date terms

Date terms are terms or phrases that represent a numbered day of the month, a month, a year, or span of years. Year terms are terms that represent a year or a span of years.

Pair type	Number of pairs	Percentage of corpus
All pairs	800	100%
Pairs with date terms in either T or H	201	25.25%
Pairs with year terms in either T or H	177	22.13%
Pairs with date terms in H	84	10.38%
Such pairs with negative entailment	39	4.88%
Pairs with year terms in H	76	9.50%
Such pairs with negative entailment	34	4.25%
Pairs with four-digit years in the H in the range 1000 to 2100.	72	9.00%
Such pairs with negative entailment	30	3.75%

We performed a manual analysis of the RTE3 development set, counting pairs that explicitly refer to a numbered day of the month, a month, a year, or a range of years (including decades and centuries) (see Table 1). Such date terms appeared in 201 pairs, 177 of which (88.1%) included a year in either H or T. Of the 201 pairs, 76 (37.8%) included a year in H, comprising 9.25% of RTE3D. Among the pairs with years in the hypothesis, 39 (51.3%) had negative entailment.

2.2 System framework

Our system precludes entailment by comparing dates in the hypothesis and texts. For each entailment pair, the system identifies dates and maps each natural-language date to a span of viable calendar dates. It then compares the overlap between the dates in the text and hypothesis sentences, predicting negative entailment according to the heuristic provided in 2.4.

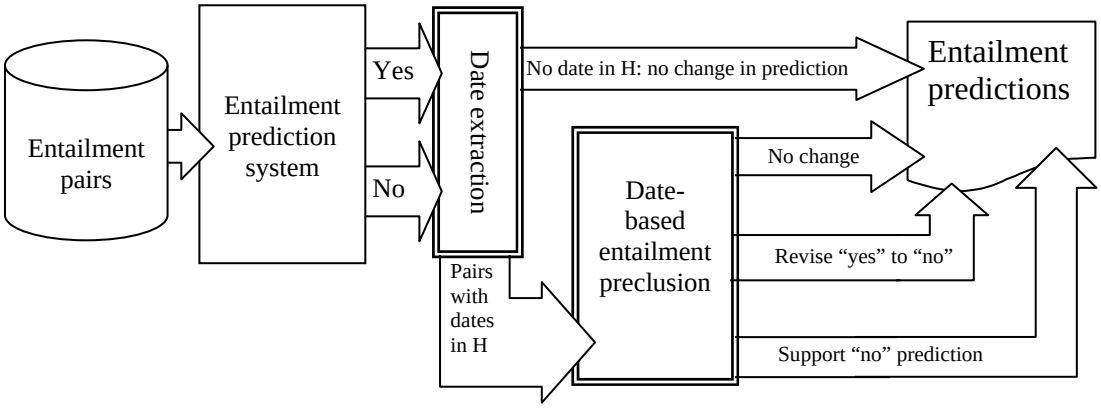

Figure 1: Possible implementation model for date-based entailment preclusion. After an entailment prediction system predicts entailment for the pairs in the corpus, this module would extract dates from the pairs. For pairs with dates in H, the system would compare the dates in H and in T, and perform one of three actions: a) pass the pair through without altering the entailment prediction; b) revise a yes prediction to a no; c) provide additional support to a no prediction. Procedures in double-walled cells are reported in this paper.

Since the system operates on a subset of entailment pairs, and only rules out entailment, we envision that our work would be embedded in a larger entailment system (see Figure 1). Our work would corroborate or reject the entailment predictions from other system components, for those pairs containing dates. The actions to be performed by our system are shown in Table 2.

We developed our system using the 800-pair development data set from the third PASCAL Recognising Textual Entailment Challenge (RTE3D) (Giampiccolo et al.,

2007). We evaluated system performance on the 800-pair test data set from RTE3 (RTE3T), with the 800-pair test data set from RTE1 (RTE1T) as a secondary evaluation set. Data was stored in an Oracle database, and the system's decision-making procedures were coded in Oracle PL/SQL.

Table 2: Actions to be performed by date-based preclusion module when evaluating entailment pairs containing dates

Predicted entailment value of pair according to previous modules	Do dates preclude entailment?	
	Yes	No
	Date-based preclusion module's action	
"YES"	Revise entailment prediction to "no"	no change
"NO"	Increase confidence in previous prediction	no change

We used version 1.6 of the Stanford parser (Klein & Manning, 2003a, 2003b) to produce a dependency parse for each sentence, and we produced a normalized word list by stripping leading and trailing nonword characters from the tokens identified by the parser. For instance, the token `(1923)` would be normalized to `1923`, and `socialism"` to `socialism`.

2.3 Date identification and extraction

The system identifies dates that contain a year. It does not identify dates with only a month or day, relative temporal references (e.g., "last year," "tomorrow"), decades (e.g., "1980s," "2000's"), or centuries (e.g., "17th century"); nor does it identify year ranges separated by hyphens (e.g., "1900-1923").

The system identifies years through a regular expression that matches all numbers between 1000 and 2100. It identifies months and days by comparing terms against

vocabularies. We created a vocabulary of month terms, which included both full month names (e.g., “August”) and abbreviations (e.g., “Aug”), and day terms, which included the cardinal numbers 1 through 31 and the ordinals 1st through 31st.

Table 3: Date extraction precedence

Date classes marked with an asterisk were not present in the development set, so we did not create code to handle them specifically. Since we developed the sequence (including the marked terms) without consulting the data set, we include them here for reference.

Date types marked “exact” indicate a single day, month, or year, without explicitly mentioning a range of dates. Date types marked “range” explicitly indicate the start and end of a range of dates.

Extraction order	Example date	Date type code	Date class
1	August 1, 2003 to September 2, 2004 *	ymd_to_ymd	Range
2	August 2003 to September 2004 *	ym_to_ym	Range
3	2003 to 2004	y_to_y	Range
4	August 1 to September 2, 2003	md-to-md_y	Range
5	August to September, 2003 *	m-to-m_y	Range
6	August 1 to 2, 2003 *	m_d-to-d_y	Range
7	August 1, 2003	m_d_y	Exact
8	August 1-2, 2003	m_d-d_y	Range
9	August 2003	m_y	Exact
10	August-September 2003	m-m_y	Range
11	2003	y	Exact

We developed a sequence to dictate the order of extraction of different types of dates. The sequence, given in Table 3, is designed to reduce the likelihood that a date range is interpreted as a specific date, and to ensure that granularity—the smallest unit of time in each date—is preserved. For example, the sequence ensures that “from May to June 2003” is not interpreted as “May 2003” or “June 2003” alone, and that “August 16, 1942” is not interpreted as “August 1942” or “1942.”

2.3.1 Ranges with multiple years (y_to_y)

The system identifies ranges with multiple years (e.g., “1811 to 1886”) by first identifying all cases where a single sentence contains at least two years. It checks each pair of years against the following conditions: the first year term must be subordinate in the dependency parse to one of the words “from”, “between” or “of”, and the second term must be subordinate to “from”, “between”, “of”, “to”, or “until”. The pair must have a superordinate term in common. The path length from each year term to the nearest common superordinate term must differ by less than 3. If these conditions are met, the two year terms are the bounds of a date range, with the sequentially first (i.e., first in sentence) year as the start date and the sequentially second year as the end date. Each pair of years is considered separately; if multiple pairs meet these criteria, the system finds multiple year ranges for that sentence.

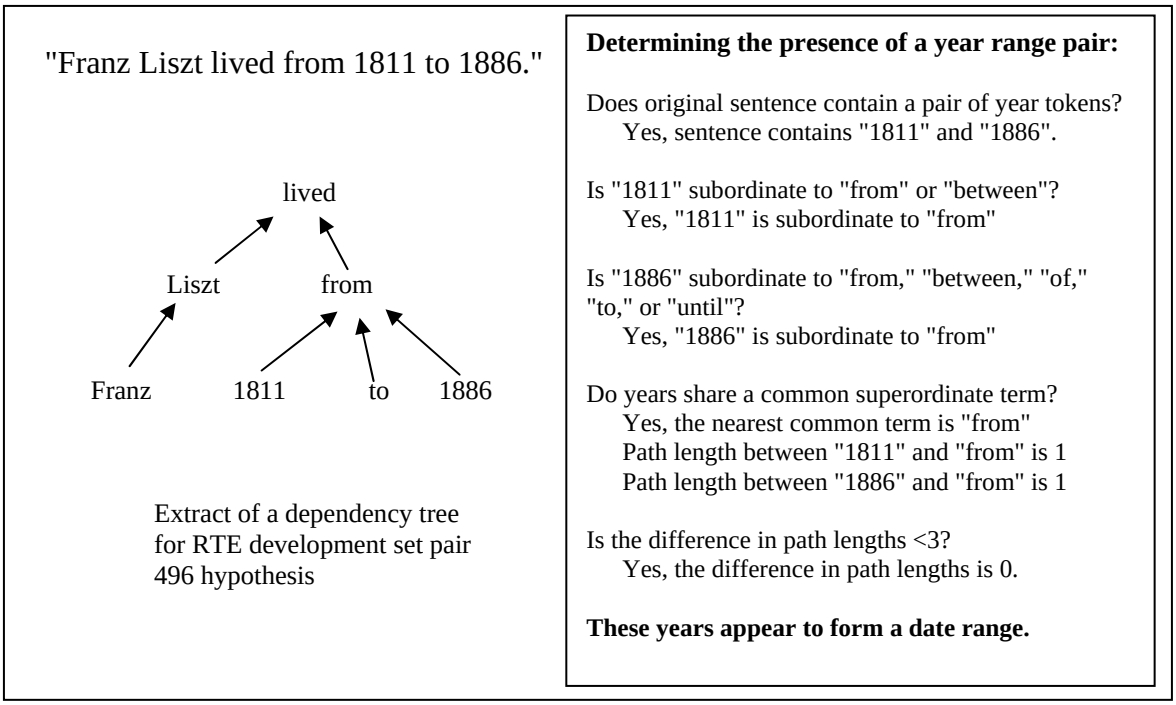

Figure 2: Example of determining if a pair of years represents a date range.

As an example, consider the hypothesis from the RTE3 development set, pair 496, shown in Figure 2. The system identifies the tokens “1811” and “1886” as years. The system checks if the sequentially first year token, “1811,” is subordinate in the dependency parse to one of the terms “from,” “between,” or “of.” "1811" is subordinate to the word “from.” The system checks if the sequentially second year term, “1886,” is subordinate to “from,” “between,” “of,” “to,” or “until.” “1886” is subordinate to “from.” The system checks if the year terms share a superordinate term. They share “from” and “lived” as superordinate terms. The distance from the first token, “1811,” to the nearest common superordinate term, “from,” is 1, as is the distance between “1886” and “from.” The difference in these distances is 0, which is less than 3. This pair represents a date range, with "1811" as the start year and "1886" as the end.

2.3.2 Ranges with multiple months and days and a single year (md-to-md_y)

For ranges with a single year and multiple months and days (e.g., “August 1 to September 2, 1993”), the system uses reasoning similar to 2.3.1 to determine if two month terms form a date range. For each such pair of month terms, the system associates the pair with the year term that most closely follows it that is within five terms following the sequentially second month term. The system searches for a day number fewer than three terms away from each month. The sequentially first month and day are taken as the beginning of the date range, and the second month and day as the end.

2.3.3 Dates and date ranges with month and year (m_d_y, m_d-d_y, m_y)

The system next identifies dates and ranges that include a single year and a single month. The system searches for month terms five or fewer terms before a year term, and matches

the nearest month to the year. Since date ranges with two months have already been identified, the likelihood of incorrectly pairing a month to the year is minimized.

These (month, year) pairings serve as candidates for three date statement classes: First, the system searches for day terms (described above) from 2 terms prior to the month term, to immediately prior to the year term. This range allows for both American-style dates (”April 1, 1992”) and European-style dates (”1 April, 1992”) to be identified, as well as expressions such as “1st of April, 1992”. (The system assumes that the month term occurs before the year term in all cases.)

For pairs where no day term is found, the system searches for ranges of days (as in “April 1-15, 1992”, “1st-10th of April, 1992”) between the month and year and one to two terms prior to the month. The system searches for these ranges separately from individual dates because the hyphenated terms are treated as single tokens in the database.

For (month, year) pairs for which no day or day range is found, the system concludes that the date refers to a month and year alone (e.g., “April 1992”).

2.3.4 Hyphenated month ranges (m-m_y)

Date ranges that include hyphenated month ranges (“April-June 1992”) are identified and matched to corresponding years in a manner similar to 2.3.3. The system searches for hyphenated month ranges within five terms immediately prior to years.

2.3.5 Single year alone (y)

The system takes any numeric token between 1000 and 2100 that does not fall into one of the above categories to be a year standing alone.

2.3.6 Converting each natural-language date into a span of calendar dates

Since dates and ranges can refer to different spans of time depending on the granularity of the dates, the system maps each date found to a span of calendar dates, by identifying a "start day" and "end day" for the date. Each date has a granularity, the smallest time unit included in the date: either "day," "month," or "year". For a date with "day" granularity—that is, where the smallest unit expressed in the date is the day of the month—the start day and end day are equal to the date itself. For example, "April 1, 1992" would have 4/1/1992 as the start day and 4/1/1992 as the end day. For a range with "day" granularity, the start date and end date of the range are the start day and end day, respectively, of the span. For an exact date or range with "month" or "year" granularity, the system uses the earliest possible day and month for the start day, and the latest possible day and month for the end day. For example, the system assigns the start day 8/1/1995 and the end day 8/31/1995 to the date "August 1995," which has "month" granularity (see further examples in Table 4). The system assumes that February has 28 days unless a date refers explicitly to February 29. Universally assuming a 29-day February would allow better matching on leap days, but Oracle's date functions do not accept a 29-day February in nonleap years.

Table 4: Date statement examples and the date spans they imply

Date as appears in text	Date class	Granularity	Start day	End day
August 5-12, 2003	range	Day	8/5/2003	8/12/2003
May to June 2004	range	Month	5/1/2004	6/30/2004
1999 to 2001	range	Year	1/1/1999	12/31/2001
2003	exact	Year	1/1/2003	12/31/2003
April 10, 1998	exact	Day	4/10/1998	4/10/1998

2.4 Entailment method

The system detects only false entailment. It compares dates in the hypothesis to those in the corresponding text, using the start and end days calculated in 2.3.6. If no date exists in the hypothesis that can be inferred from a date in the text, the system precludes entailment.

We define the following pairwise relationships on dates: Two dates D_1 and D_2 are **equal** if their spans have the same start and end days, regardless of the granularity of the original dates; D_1 and D_2 **overlap** if either the start day or end day of D_1 (but not both) falls between the start and end days of D_2; they are **disjoint** if the end day of D_1 precedes the start day of D_2; and they have a **superset/subset** relationship if D_2 (the **subset**) has both its start and end days (non-strictly) between the start and end days of D_1 (the **superset**) and the two dates are not equal (see Figure 3 for a visual representation of these relationships).

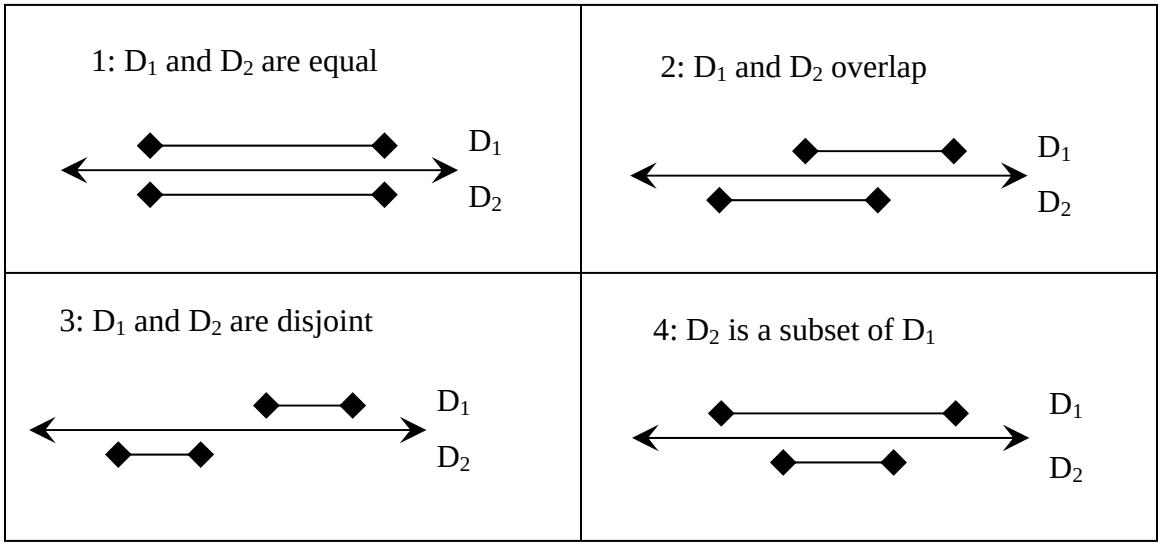

Figure 3: Visual representation of date relationships

For each sentence pair, the system compares each date in the hypothesis D(H) to each date in the text D(T). The system will rule out preclusion for a sentence pair if *any* (D(H),D(T)) meets one of the heuristics ruling out preclusion. If no such combination exists, *and* (D(H),D(T)) exists that meets one of the heuristics precluding entailment, then entailment is precluded for the sentence pair. Comparisons are based on the equality, overlapping, disjunction, and superset/subset definitions above, in addition to the classification of the date as "exact" or "range" as described in Table 3. Exact dates specify a single day, month, or year, without explicitly indicating a span of time. Range dates include start and end years, months, or days, to explicitly indicate duration.

Preclusion is ruled out for any sentence pair, and we will say that D(H) and D(T) align, where D(T) and D(H) exist such that any of the following conditions is met:

1. D(H) equals D(T) and both D(H) and D(T) are of the same date type ("range" or "exact")
2. D(T) is a subset of D(H), and D(H) is exact
3. D(T) is a subset of D(H), both D(H) and D(T) are ranges, and D(H) is less granular than D(T), with start and end dates in corresponding months and/or years

Entailment is precluded for any sentence pair where preclusion is not ruled out above, *and* D(H) and D(T) exist such that either:

4. D(H) and D(T) overlap, or D(H) and D(T) are disjoint
5. D(H) is a subset of D(T)

Two of these conditions warrant further explanation:

- Condition 2 identifies cases where an exact date D(H) (with granularity of month or year) contains a range D(T):
    - D(T): "May 15-24, 2003" D(H): "May, 2003"
    - D(T): "April 10, 2003" D(H): "April 2003"
    - D(T): "April-June, 2005" D(H): "2005"

- Condition 3 rules out preclusion when D(H) is a less granular expression of D(T).
    - D(T): "Apr 10, 2002 to Jun 23, 2003" D(H): "April 2002 to May 2003"
    - D(T): "Jun 1996 to Sept 1999" D(H): "1996 to 1999"

3 Results

The system analyzed three data sets: RTE3D, RTE1T, and RTE3T (see section 1.5, page 10), each with 800 pairs and approximately 50% positive entailment and 50% negative entailment. Our system was developed and optimized using RTE3D, with RTE3T set as the primary evaluation set. We analyzed RTE1T to measure performance outside of the RTE3 Challenge environment.

For each set, we show a manual count of pairs with dates; we examine the performance of the year extraction method (see section 2.3, page 14); and we discuss the results of the entailment preclusion method (see section 2.4, page 20).

In the analyses below, the system performed “correctly” if it predicted negative entailment for a pair where the hypothesis is not entailed by the text. It performed “incorrectly,” or produced a false negative, if it predicted negative entailment for a pair that had positive entailment. It “missed” the entailment relationship if it came to no conclusion on a negative pair.

3.1 RTE3 Development

See section 2.1, page 11, for a manual count of pairs with dates in RTE3D.

Table 6: Text excerpts of false positives and negatives from year identification in RTE3D
Excerpts labeled "T" are from texts; those labeled "H" are from hypotheses.

Pair ID	T/H	False positives
228	T	"…which is **1800** miles to the east of Great Britain…"
326	T	"…half the annual **1200** quota…"
342	T	"…thought to have died in battle more than **2000** years ago…"
392	T	"…12:05 EDT (**1605** GMT)…"
500	T	"…**2500** years after it was written…"
532	T	"…with nearly **1000** people…"
695	T	"…traveled **2000** miles…"
Pair ID	**T/H**	**False negatives**
463	T	"…king of the Chinese State of Qin from **247** BCE to **221** BCE… emperor of a unified China from **221** BCE to **210** BCE…"
499	T	"Aeschylus was born in **525** BC…"
537	T	"…built around **280** BC…"
538	T	"…built around **280** BC…"
538	H	"The Pharos lighthouse was built around **280** BC."

In RTE3D, the year identification pattern matched 338 years (true positives), while incorrectly identifying 7 nonyear tokens as years (false positives), for a precision of 338/345 (98.0%). The regular expression failed to match 8 years in five sentences (false negatives), since these years had only three digits. Recall for identifying years was 338/346, or 97.7%. False positives and negatives are shown in Table 5. This analysis does not consider decades, centuries, or hyphenated date ranges, of which there were 17.

Table 5: Confusion matrix for system results on RTE3D
Pairs included in table are those with years in the range 1000-2100 in the hypothesis.

	This system		
Actual entailment	No prediction	Entailment precluded	Totals
YES	40	2	42
NO	13	17	30
Totals	53	19	72

The system precluded entailment for 19 pairs, of which 17 (89.5%) had negative entailment. The 17 correct pairs represent 2.1% of the corpus. The system's performance is summarized in Table 6. Of the correct pairs, 1 was of the IE task type, 3 were IR, 2 were SUM, and 11 were QA.

Table 7: Preclusion failures in RTE3D

Reason for failure	Number of pairs
Date comparison alone not sufficient: ideas from T are incorrectly associated with date in H	5
Date comparison alone not sufficient : ideas not present in T are associated with date in H	6
Incomplete reasoning: No rejection criteria defined for combination of date types in T and H	2
TOTAL	**13**

The system missed 13 pairs with a year in the range 1000-2100 in the hypothesis that should have been precluded (see Table 7). In 11 cases, date comparison alone was not enough to preclude entailment: the dates in H and T align (see section 2.4, page 20) but entailment does not hold because event details are incorrectly paired with dates in H. For example:

```
RTE3 Development Set pair 25
T: Bountiful arrived after war's end, sailing into San Francisco
      Bay 21 August 1945. Bountiful was then assigned as hospital
      ship at Yokosuka, Japan, departing San Francisco 1 November
      1945.
H: Bountiful reached San Francisco on 1 November 1945.

RTE3 Development Set pair 77
T: Following the Declaration of the Establishment of the State of
      Israel, May 14, 1948, seven Arab states entered Palestine
      and engaged Israeli forces.
H: Israeli forces attacked seven Arab states in 1948.

RTE3 Development Set pair 265
T: Fujimori charged that on January 26, 1995, Ecuador fired the
      first shot, an allegation denied by Ecuador's leader, Sixto
      Duran-Ballen. Predictably, each side blamed the other for
      starting the 1995 conflict, just as each pointed the finger
```

```
      of guilt to the other for provoking the border war of 1941,
      when Peru took most of the 120000 square miles in
      contention between the two countries.
H: President Fujimori was re-elected in 1995.
```

In pair 25, the date in H aligns with a date in T, but H's date is associated with a different event from T. In pair 77, the dates align but the subject and object of the associated ideas are swapped between T and H. In five cases, dates were incorrectly matched with event details in this manner. In pair 265, the date in H aligns with two dates from T, but an idea in H ("re-election") does not appear at all in T. In six cases, hypothesis dates were matched with ideas not present in the text.

In 2 cases, the system did not draw a conclusion on entailment because the entailment method neither precluded entailment nor ruled out preclusion:

```
RTE3 Development Set Pair 494:
T: Victor Emmanuel III yielded most of his powers to his son
      Umberto II in 1944, when Umberto was appointed as
      Lieutenant General of the Realm, and finally abdicated in
      1946.
H: Victor Emmanuel III was king of Italy from 1900 to 1946.

RTE3 Development Set Pair 497:
T: The Altenburg is located in the Jenaer street which lies just
      outside of Weimar's city centre beyond the Ilm park. It was
      built in 1810-1811 and during the years following 1848,
      Princess Carolyne of Sayn Wittgenstein and her husband,
      Franz Liszt, lived there.
H: Franz Liszt lived from 1811 to 1886.
```

In pair 494, the dates in T are exact and the date in H is a range. For both T dates, D(T) is a subset of D(H). The method to preclude entailment makes no judgment in this case, so entailment is not precluded. In pair 497, the first date in T is not interpreted, because it is a hyphenated year range. The second date in T is a subset of the range in H. Once again, D(T) is a subset of D(H) and no judgment is made.

In four pairs, no decision was made because the system does not decipher decade statements (e.g., “1990s”).

The system produced false negatives in two pairs:

```
RTE3 Development Set pair 495
T: Victor Emmanuel III, 1869-1947, king of Italy (1900-1946),
      emperor of Ethiopia (1936-43), king of Albania (1939-43),
      son and successor of Humbert I.
H: Victor Emmanuel III was king of Italy from 1900 to 1946.

RTE3 Development Set pair 655
T: Argentina was still obsessed with the Falkland islands even in
      1994, 12 years after its defeat in the 74-day war with
      Britain.
H: The Falklands War took place in 1982.
```

In pair 495, the system does not interpret the hyphenated year ranges. For pair 655, the system recognizes the date in the text, but it does not interpret “1994” and “12 years after” as a reference to the year 1982.

3.2 RTE1 Test

A manual inspection of RTE1T (see Table 8) reveals that 173 pairs contain explicit references to days of the month, months, years, decades, or centuries. Of these pairs, 140 (80.9%) refer to a year. Of the 173 pairs, 68 (39.3%) include a year in the hypothesis, comprising 8.5% of RTE1T. Of the 68 pairs with years in the hypothesis, 32 (47.1%) have negative entailment.

The system’s year-matching pattern correctly identified 198 years and incorrectly identified 7 nonyear tokens as a year, for a precision of 198/205 (96.6%). It failed to identify 2 years that fell outside of the pattern’s search range (both references to “AD 50”), for a recall of 198/200 (99.0%).

Table 9: Pairs in RTE1 test set with date terms

Date terms are terms or phrases that represent a numbered day of the month, a month, a year, or span of years. Year terms are terms that represent a year or a span of years.

Pair type	Number of pairs	Percentage of corpus
All pairs	800	100%
Pairs with date terms in either T or H	173	21.63%
Pairs with year terms in either T or H	140	17.50%
Pairs with date terms in H	83	10.38%
Such pairs with negative entailment	39	4.88%
Pairs with year terms in H	68	8.50%
Such pairs with negative entailment	32	4.00%
Pairs with four-digit years in H in the range 1000-2100	59	7.38%
Such pairs with negative entailment	27	3.38%

The system precluded entailment for 11 pairs, doing so correctly for 9 pairs (81.8%). The 9 correct pairs represent 1.1% of the corpus. In 2 pairs, nonyear strings were identified as years, although the system correctly precluded entailment. The system's performance is summarized in Table 9.

Table 8: Confusion matrix for system results on RTE1T

Pairs included in table are those with years in the range 1000-2100 in the hypothesis.

	This system		
Actual entailment	No prediction	Entailment precluded	Totals
YES	31	2	33
NO	20	7	27
Totals	51	9	60

The system missed precluding entailment for 20 pairs with a year in the hypothesis (see Table 10). In 2 cases, date comparison alone was not sufficient because a key word or phrase from T was removed in H. For example:

Table 10: Preclusion failures in RTE1T

Reason for failure	Number of pairs
Date comparison alone not sufficient: ideas from T are incorrectly associated with date in H	11
Date comparison alone not sufficient : ideas not present in T are associated with date in H	6
Date comparison alone not sufficient: modifying phrase absent from H changes meaning of idea from T	2
Data error: Missing T data in database	1
TOTAL	**20**

```
RTE1 Test Set pair 2082:
T: Microsoft was established in Italy in 1985.
H: Microsoft was established in 1985.
```

Here, the dates in T and H match, but the omission of the phrase “in Italy” from H alters the meaning of the sentence, and entailment does not hold.

Entailment was not precluded for one pair because the pair’s T data was omitted from the database. In two pairs, no entailment decision was made because the system does not decipher decade statements.

The system produced false negatives in 2 pairs:

```
RTE1 Test Set pair 1214
T: It is planned that by the end of the year 2001, France will
      have minted 7.6 billion Euro coins weighing 30 thousand
      tons or approximately four times the equivalent weight of
      the Eiffel Tower.
H: According to plans, France should have minted 7.6 billion Euro
      coins before 2002.

RTE1 Test Set pair 1877
T: The English team arrived last night in Lisbon, Portugal, to
      play its first Euro 2004’s match.
H: Euro 2004 is held in Portugal.
```

In pair 1214, the date range indicated by the hypothesis, “before 2002,” can include any year up to 2001 (i.e., a range that could be expressed “1/1/0001 – 12/31/2001”), which is

equal to the range implied by the text's "by the end of the year 2001." The system does not interpret the modifiers "by the end of" and "before," and identifies the time spans for the text and hypothesis as 1/1/2001-12/31/2001 and 1/1/2002-12/31/2002 respectively. In pair 1877, the date in the text is not recognized because the system only identifies year terms that are exactly four characters long.

3.3 RTE3 Test

A manual inspection of RTE3T (see Table 11) reveals that 239 pairs contain explicit references to days of the month, months, years, decades, or centuries. 223 of these pairs (93.3%) refer to a year, of which 66 (29.6%) of the 223 pairs include a year in the hypothesis, comprising 8.25% of RTE3T. Of the 67 pairs with date terms in the hypothesis, 27 (40.3%) have negative entailment.

Table 11: Pairs in RTE3 with date terms

Date terms are terms or phrases that represent a numbered day of the month, a month, a year, or span of years. Year terms are terms that represent a year or a span of years.

Pair type	Number of pairs	Percentage of corpus
All pairs	800	100%
Pairs with date terms in either T or H	239	29.88%
Pairs with year terms in either T or H	223	27.88%
Pairs with date terms in H	67	8.38%
Such pairs with negative entailment	27	3.38%
Pairs with year terms in H	66	8.25%
Such pairs with negative entailment	27	3.38%
Pairs with four-digit years in H in the range 1000-2100	63	7.88%
Such pairs with negative entailment	26	3.25%

Table 12: Confusion matrix for system results on RTE3T
Pairs included in table are those with years in the range 1000-2100 in the hypothesis.

	This system		
Actual entailment	No prediction	Entailment precluded	Totals
YES	27	3	30
NO	24	9	33
Totals	51	12	63

The system's year-matching pattern correctly identified 336 years and incorrectly identified 1 nonyear token as a year, for a precision of 336/337 (99.7%). It failed to identify 3 years that fell outside of the pattern's search range (all with three digits), for a recall of 336/339 (99.1%).

The system precluded entailment for 12 pairs, doing so correctly for 9 pairs (75%). The 9 correct pairs represent 1.13% of the corpus. Of these pairs, 7 were in the QA problem class, 1 in the SUM class, and 1 in the IR class. The system's performance is summarized in Table 12.

The system missed precluding entailment for 17 pairs with dates in the range 1000 to 2100, summarized in Table 13. In one pair, the system was not able to identify a pair of years as a date range. The system finds year pairs in the formats "from XXXX to YYYY," "of XXXX to YYYY," and "between XXXX and YYYY." In RTE3 test set pair 747, the date range in the text was expressed, "began in 1982 and continued through 1994," and was not identified as a range. In two pairs, the event stated in H matched an event in T, but in the hypothesis the antonym of the original verb was provided. In 2 cases, the system did not draw a conclusion on entailment because its reasoning neither precluded entailment nor ruled out preclusion.

Table 13: Preclusion failures in RTE3T

Reason for failure	Number of pairs
Date comparison alone not sufficient: ideas from T are incorrectly associated with date in H	5
Date comparison alone not sufficient : ideas not present in T are associated with date in H	6
Date comparison alone not sufficient: Modifying phrase present in H changes meaning of idea from T	1
Date comparison alone not sufficient: Verb in H antonymous to verb in T	2
Incomplete reasoning: No rejection criteria defined for combination of date types in T and H	2
Incomplete reasoning: Date range in T expressed in a manner not recognized by date extraction algorithm	1
TOTAL	**17**

The system produced false negatives in 3 pairs:

```
RTE3 Test Set pair 486
T: Charles de Gaulle, 1890–1970, French general and statesman,
      was the first president (1959–69) of the Fifth Republic.
H: Charles de Gaulle died in 1970.

RTE3 Test Set pair 503
T: The reigns of Victor Emmanuel II (1861–78) and Humbert I
      (1878–1900), and the first half of the reign of Victor
      Emmanuel III (1900–1946) were marked by moderate social and
      political reforms and by some industrial expansion in
      Northern Italy (mainly in the 20th cent.).
H: Victor Emmanuel III was king of Italy from 1900 to 1946.

RTE3 Test Set pair 615
T: The world’s population is set to reach a staggering 10bn by
      the middle of 21st century up from 5.7bn now.
H: The world population will probably reach over 10 billion in
      2050.
```

In pairs 486 and 503, the system did not interpret hyphenated date ranges. In pair 615, the system did not provide a year value for “the 21st century”, and would have been unable to equate the phrase “the middle of” to a specific date therein.

Table 14: Confusion matrix for system results on the combined corpus

Actual entailment	This system		Totals
	No prediction	Entailment precluded	
YES	116	7	123
NO	50	33	83
Totals	166	40	206

3.4 Discussion

On the combined (RTE3D + RTE1T + RTE3T) corpus, the system identified 887 terms as years, of which 872 were identified correctly (98.3% precision). There were 885 years in the corpus (not counting decades, hyphenated year ranges, or centuries) (98.5% recall). Across the combined corpus, dates with years appear in 540 pairs.

The system precluded entailment for 42 pairs in the combined corpus, and entailment did not hold for 35 of those pairs (83.3%). See Table 14 for a summary of the system's performance. It was unable to preclude entailment for 50 pairs with years in the range 1000-2100 in the hypothesis; the causes of preclusion failure are summarized in Table 15, can be grouped into two categories: cases of data loading errors or incomplete date comparison reasoning (7 pairs), and pairs where date comparisons alone were not sufficient to preclude entailment (44 pairs).

Table 15: Causes of preclusion failure in combined corpus

Reason for failure	Number of cases	Percentage of failures	Percentage of pairs with entailment=no and years in 1000-2100
Date comparison alone not sufficient	44	88.0%	53.0%
Incomplete reasoning or data error	6	12.0%	7.2%

3.5 Future work

Across the three data sets (2400 pairs total, 83 pairs with years in the range 1000-2100 in the hypothesis), the system failed to preclude entailment for 50 pairs. For 4 pairs, the system did not predict false entailment because it did not compare the dates in those pairs. These cases can be corrected by providing additional date comparison steps. In 1 pair, a data loading error prevented the system from precluding entailment. In 1 pair, a date range was expressed in a fashion not recognized by the system. In the 44 remaining missed pairs, entailment was not precluded because date comparison alone was not sufficient to do so; entailment preclusion is unattainable for these pairs without expanding the scope of the system beyond date comparison.

For three pairs, the system produced false negatives because it did not interpret hyphenated dates. Additional date extraction steps can improve performance on these pairs. For six pairs containing decade statements, no prediction on entailment was reached. Interpretation of prepositions and prepositional phrases modifying temporal statements, present in three false negatives, and interpretation of century statements, present in one false negative, would have further improved performance. The system has been designed for the RTE challenges and may need modification to achieve maximum effectiveness in another context.

In light of the error analysis provided above, we make two suggestions. First, extend the current system to include decade and century handling, deciphering of hyphenated date ranges, handling dates that do not include year, and the interpretation of temporal prepositional phrases and other temporal modifiers. This will provide an increase in system performance without expanding the scope of the system beyond its current

framework. Second and more significantly, expand this system to identify events and to associate events with their dates of occurrence. Such a system may be able to preclude entailment on the 44 pairs outlined above where date comparison alone is not sufficient to preclude entailment. Such a system may also be able to provide support for positive entailment pairs with temporal information.

3.6 Summary of results

In RTE3D, the system identified 345 strings as years and did so correctly 338 times, for a year-identification accuracy of 98.0%. There were 346 years in the data set, for a year-identification recall of 97.7%. The system was 89.5% accurate in precluding entailment, predicting correctly in 17 of the 19 pairs for which it made a prediction. The system correctly precluded entailment for 2.1% of the 800 pairs in RTE3D.

In RTE1T, the system identified 205 strings as years and was correct 198 times, for a year-identification accuracy of 96.6%. There were 200 years in the data set, for a year-identification recall of 99.0%. The system precluded entailment in 11 pairs, doing so correctly in 9, for an accuracy of 81.8%. The system correctly precluded entailment for 1.1% of 800 pairs.

In RTE3T, the system identified 337 strings as years and was correct 336 times, for a year-identification accuracy of 99.7%. There were 339 years in the data set, for a year-identification recall of 99.1%. The system correctly precluded entailment in 9 out of the 12 pairs for which it made a prediction (75% accuracy). The system made the correct judgment in 1.1% of 800 pairs.

In aggregate (considering RTE3D, RTE1T, and RTE3T as a combined corpus), the system identified 887 strings as years and was correct 872 times, for an accuracy of 98.3%. There were 885 years in the combined corpus, for a recall of 98.5%. The system was correct in precluding entailment in 35 out of the 41 pairs for which it made a prediction (85.3% accuracy). The system correctly precluded entailment for 1.5% of 2400 pairs.

To examine to what extent this system improves an existing system, we compared the performance of a baseline system (Blake, 2007) on RTE3D and RTE3T to the performance it would have achieved with the addition of our method. The baseline system predicts positive entailment by using a decision tree to compare grammatical features in the H and T sentences. It identifies cases where the subject, verb, and object of H sentences match those in T sentences, and predicts positive entailment for those pairs.

Table 16: Performance of baseline system alone, and with date-based preclusion method
Only pairs with years in the hypothesis in the range 1000-2100 are considered.

RTE3D
Baseline alone

	Predicted		
Actual	YES	NO	Totals
YES	21	21	42
NO	18	12	30
Totals	39	33	72

RTE3D
Baseline with preclusion method

	Predicted		
Actual	YES	NO	Totals
YES	20	22	42
NO	8	22	30
Totals	28	44	72

RTE3T
Baseline alone

	Predicted		
Actual	YES	NO	Totals
YES	28	9	37
NO	11	15	26
Totals	39	24	63

RTE3T
Baseline with preclusion method

	Predicted		
Actual	YES	NO	Totals
YES	28	9	37
NO	10	16	26
Totals	38	25	63

Among sentences with years in the range 1000 to 2100 in the hypothesis, the baseline system had an accuracy of 33/72 (45.8%) in RTE3D and 43/63 (68.3%) in RTE3T. When combined with our method, the scores for the system improved to 42/72 (58.3%) for RTE3D and 44/63 (69.8%) in RTE3T (see Table 16). For the full 800-pair sets, inclusion of our method improved accuracy from 399/800 (49.9%) to 408/800 (51.0%) in RTE3D, and from 527/800 (65.9%) to 528/800 (66.0%) in RTE3T.

4 Conclusion

We have presented a system that precludes entailment by identifying dates in the hypothesis that do not correspond to dates in the text. The system requires that sentences be Stanford-parsed, with tokens normalized. The system operates independently of other efforts to resolve entailment, and we envision that it would be part of a larger entailment resolution system.

Our method first identifies and extract dates from natural language text, then precludes entailment by comparing the dates in the text and hypothesis sentences.

We proposed a precedence for extracting dates that limits the chance that components of a date (such as months, days, or portions of a date range) will be overlooked (see Table 3). We introduced four relationships between two dates: equality, overlap, disjunction, and superset/subset; and a method that considers both the relationship between dates and the type of date (exact or range) to preclude entailment (see section 2.4). Finally, we have conducted an extensive failure analysis of our system on three data sets from the RTE Challenges. This analysis suggests that, although temporal features alone improve entailment accuracy, the combination of entailment features and event recognition is ultimately required to accurately predict entailment for pairs with temporal information.

Scores in the RTE challenges are closely clustered: fewer than seven percentage points separate the accuracy scores of the third-best performing system and the thirteenth-best in RTE3 (Giampiccolo et al., 2007). The system presented here correctly precluded

entailment for between 1.1% and 2.1% of each test set it processed. Our method can contribute to a larger entailment system by precluding entailment on these pairs or providing support to predictions made by other components. To better understand our method's contribution, we propose that designers integrate our method into such an entailment system along with other components, and carefully evaluate the contributions of each component.

Temporal information is an important area for research in entailment and in other text applications. Our work adds to the available knowledge on temporal features by providing a means to identify dates and by exploring the roles that they play in entailment. There is much research to be done in this area, and we encourage researchers to explore temporal features further, both within and beyond the realm of textual entailment.

5 Acknowledgments

Thanks to Cathy Blake for advising this paper, for dedicating time and effort to helping me produce a well-designed system and report, for providing material assistance by way of database space and preprocessing, for getting me interested in text mining (and teaching me pretty much everything I know), and for never allowing me to be satisfied with the merely adequate.

Thanks to the faculty and staff of UNC-SILS for making this educational experience a possibility and a success.

Thanks to Amol Bapat for being sporting in dealing with my data nightmares.

Thanks to Mom and Dad for being Mom and Dad, and for encouraging me to learn at every opportunity.

6 Appendix: PL/SQL script

In the code below, references are made to the following tables. A brief description of each table's schema is provided:

dre_months(monthterm, monthnum, mtype, numdays)

This table holds the names, numbers, and number of days in each month. **monthterm** holds the token that represents the month name. **monthnum** contains the number of the month (e.g., October = 10). **mtype** is "full" if the month term is the full name of the month or "abbr" if it is an abbreviation. **numdays** is the number of days in each month. February is assumed to always have 28 days.

dre_numbers(numterm, numval)

This table holds the tokens that represent numbers that are valid day values for a month, from 1 through 31. **numterm** is the token that represents the number—either cardinal or ordinal—and **numval** is the corresponding numerical value.

dre_numranges(numrange, numval1, numval2)

This table contains all the possible number range values that could represent valid date ranges within a month (e.g., "1st-15th," "10-12"). **numrange** is the token representing the number range, **numval1** is the value of the starting number in the token, and **numval2** is the value of the ending number.

dre_monthranges(monthrange, mtype, monthnum1, monthnum2, month2days)

This table contains all the possible month ranges that could exist within a year. **monthrange** contains the month range tokens (e.g., “January-April”), **mtype** is “full” or “abbr” depending on whether the month terms composing the month range token are full names or abbreviated, **monthnum1** and **monthnum2** are the number of the two months, and **month2days** is the number of days in the ending month.

rte3dtext(pairid, usentid, source, sentid, text)

This table contains the text of the data set to be analyzed, named thus because the example code provided is for the RTE3 development data set. Each row of the table contains one sentence from the corpus. **pairid** is the unique pair id provided in the corpus for the entailment pair, **usentid** is a unique number provided to each sentence in the table, and **source** is ‘E’ if the sentence is from the text and ‘H’ if the sentence is from the hypothesis. **sentid** indicates the ordering of sentences, starting from zero, for each sentence with the same (pairid, source), and **text** is the text of the sentence.

rte3d_nwd(usentid, termid, term)

This table contains the normalized words for each word in the corpus. **usentid** is the unique sentence ID (from rte3dtext) of the sentence containing the listed word. **termid** is the position of the word in the sentence, starting with 1. **term** is the word token itself, extracted from the text of the sentence.

rte3dstan(pairid, usentid, source, sentid, stantype, term1, termid1, term2, termid2)

This is the table of Stanford Parser dependency relationships between terms in each sentence. **pairid**, **usentid**, **source**, and **sentid** are as in rte3dtext. **stantype** is the Stanford Parser dependency type that exists between the two named terms. **term1** is the non-normalized token representing the first (superordinate) term in the dependency, and **termid1** is the termid of that token (as in rte3d_nwd). **term2** is the token for the second (subordinate) term in the dependency, and **termid2** is the termid of that token.

Results of the analysis are found in the table dre_rte3d_results(pairid, entailment), with **pairid** indicating the unique pair identifier for which entailment has been determined, and **entailment** indicating the entailment determination in YES/NO terms (which will always be NO for this system).

6.1 PL/SQL Script

```
-- TABLES NOT TO DELETE:
-- dre_months
-- dre_numbers
-- dre_numranges
-- dre_monthranges

-- Common tasks:

-- Drop all the tables

DROP TABLE dre_rte3d_final;
DROP TABLE dre_rte3d_candidates;
DROP TABLE dre_rte3d_allpaths;
DROP TABLE dre_rte3d_verbs;
DROP TABLE dre_rte3d_numterms;
DROP TABLE dre_rte3d_yearpairs_candidates;
DROP TABLE dre_rte3d_yearpairs;
DROP TABLE dre_rte3d_yearpairs_terms;
DROP TABLE dre_rte3d_monthpair_candidates;
DROP TABLE dre_rte3d_monthpairs;
DROP TABLE dre_rte3d_monthpairs_years;
DROP TABLE dre_rte3d_monthpairs_mdy;
DROP TABLE dre_rte3d_monthyear;
DROP TABLE dre_rte3d_my_verb_candidates;
DROP TABLE dre_rte3d_my_verbs;
DROP TABLE dre_rte3d_myr_candidates;
DROP TABLE dre_rte3d_year_candidates;
DROP TABLE dre_rte3d_final_clean;
```

```
DROP TABLE dre_rte3d_final_date;
DROP TABLE dre_rte3d_final_e;
DROP TABLE dre_rte3d_final_h;
DROP TABLE dre_rte3d_results;
DROP TABLE dre_rte3d_final_e_class;
DROP TABLE dre_rte3d_final_h_class;
DROP TABLE dre_rte3d_h_contain_e_exact;
DROP TABLE dre_rte3d_h_equal_e_exact;
DROP TABLE dre_rte3d_h_equal_e_range;
DROP TABLE dre_rte3d_exact_contains_range;
DROP TABLE dre_rte3d_mrange_cont_mdrange;
DROP TABLE dre_rte3d_h_outside_e;
DROP TABLE dre_rte3d_h_partial_e;
DROP TABLE dre_rte3d_e_contains_h;

-- Set up storage table.
CREATE TABLE dre_rte3d_final
(usentid    number,
nounid            number,
noun        varchar2(100),
verbid            number,
verb        varchar2(100),
startmonth  varchar2(100),
startday    varchar2(100),
startyear   varchar2(100),
startyearid number,
endmonth    varchar2(100),
endday            varchar2(100),
endyear           varchar2(100),
endyearid   number,
datetype    varchar2(100));


-- Find year terms
-- dre_rte3d_candidates

CREATE TABLE dre_rte3d_candidates AS
SELECT * FROM rte3d_nwd
WHERE
((REGEXP_LIKE(term, '1[0-9][0-9][0-9]') AND length(term) = 4)
OR
(REGEXP_LIKE(term, '2[01][0-9][0-9]') AND length(term) = 4)
)
;

-- Build allpaths for these sentences
-- dre_rte3d_allpaths

CREATE TABLE dre_rte3d_allpaths AS
SELECT usentid,
  CONNECT_BY_ROOT termid1 AS rootid,
  CONNECT_BY_ROOT term1 AS rootterm,
  termid2 AS leafid,
  term2 AS leafterm,
  LEVEL as pathlength,
  CONNECT_BY_ROOT term1 || SYS_CONNECT_BY_PATH(term2, '>') AS termpath,
```

```
  CONNECT_BY_ROOT termid1 || SYS_CONNECT_BY_PATH(termid2, '>') AS
termidpath,
  SYS_CONNECT_BY_PATH(stantype, '>') AS typepath
FROM rte3dstan
WHERE usentid IN (SELECT usentid FROM dre_rte3d_candidates)
CONNECT BY usentid = PRIOR usentid AND termid1 = PRIOR termid2
;

-- Build verbs for these sentences
-- dre_rte3d_verbs

CREATE TABLE dre_rte3d_verbs AS
SELECT a.usentid, a.termid1 AS verbid, b.term AS verb, max(verbness) AS
verbness FROM
(SELECT * FROM
(SELECT
  usentid,
  termid1,
  term1,
  SUM(CASE when stantype = 'nsubj' then 1 else 0 end) +
  SUM(CASE when stantype = 'dobj' then 1 else 0 end) +
  SUM(CASE when stantype = 'nsubjpass' then 1 else 0 end) verbness
FROM rte3dstan WHERE usentid IN (SELECT usentid FROM
dre_rte3d_candidates)
GROUP BY usentid, termid1, term1
ORDER BY usentid)
WHERE verbness > 0

UNION

SELECT *
FROM
(SELECT
  usentid,
  termid1,
  term1,
  SUM(CASE when stantype = 'poss' then 1 else 0 end) +
  SUM(CASE when stantype = 'prep' then 1 else 0 end) verbness
FROM rte3dstan WHERE usentid IN (SELECT usentid FROM
dre_rte3d_candidates)
AND term1 IN (SELECT term FROM lragr1 WHERE agr LIKE 'infinitive')
GROUP BY usentid, termid1, term1
ORDER BY usentid)
WHERE verbness > 1

UNION

SELECT usentid, termid1, term1, 2 AS verbness
FROM rte3dstan
WHERE usentid IN (SELECT usentid FROM dre_rte3d_candidates)
AND stantype = 'tmod') a
JOIN rte3d_nwd b ON a.usentid = b.usentid AND a.termid1 = b.termid
GROUP BY a.usentid, a.termid1, b.term
;

INSERT INTO dre_rte3d_verbs
SELECT usentid, termid, term, 0 AS verbness
```

```
FROM rte3d_nwd
WHERE usentid NOT IN (SELECT usentid FROM dre_rte3d_verbs)
AND termid = 1
;


-- Find number terms and correspond with integer value.
-- dre_rte3d_numterms

CREATE TABLE dre_rte3d_numterms AS
SELECT a.*, b.numval
FROM rte3d_nwd a
JOIN dre_numbers b ON a.term = b.numterm
;

-- Category tasks:
-- Final schema for all category tables should be:
-- usentid, nounid, noun, verbid, verb, startmonth, startday,
startyear, endmonth, endday, endyear

-- Find year pair candidate sentences. Only gets pairs that have a
"from" or "between"
-- in them, presumably indicating a year announcement.
-- dre_rte3d_yearpairs_candidates

CREATE TABLE dre_rte3d_yearpairs_candidates AS
SELECT a.usentid,
      a.rootid,
      a.rootterm,
      a.leafid AS leafid1,
      b.leafid AS leafid2,
      a.leafterm AS leafterm1,
      b.leafterm AS leafterm2,
      a.termpath AS path1,
      b.termpath AS path2,
      (a.pathlength + b.pathlength)/2 AS pathavg,
      abs(a.pathlength - b.pathlength) AS pathdisc
FROM dre_rte3d_allpaths a JOIN dre_rte3d_allpaths b ON a.usentid =
b.usentid AND a.rootid = b.rootid
WHERE (a.usentid, a.leafid) IN (SELECT usentid, termid FROM
dre_rte3d_candidates)
AND (b.usentid, b.leafid) IN (SELECT usentid, termid FROM
dre_rte3d_candidates)
AND a.leafterm < b.leafterm
AND abs(a.pathlength - b.pathlength) < 3
AND (a.usentid, a.rootid) IN (SELECT usentid, verbid FROM
dre_rte3d_verbs)
AND (LOWER(a.termpath) LIKE '%of>%' OR LOWER(a.termpath) LIKE '%from>%'
OR LOWER(a.termpath) LIKE '%between>%')
AND (LOWER(b.termpath) LIKE '%of>%' OR LOWER(b.termpath) LIKE '%from>%'
OR LOWER(b.termpath) LIKE '%between>%' OR LOWER(b.termpath) LIKE
'%to>%' OR LOWER(b.termpath) LIKE '%until>%')
ORDER BY a.usentid, a.rootid
;

-- Get the candidate path that has the shortest route between the verb
and the year terms
```

```
-- These paths will determine the year-verb pairings for future
building.

CREATE TABLE dre_rte3d_yearpairs AS
SELECT * FROM dre_rte3d_yearpairs_candidates
WHERE (usentid, leafid1, leafid2, pathavg) IN
      (SELECT usentid, leafid1, leafid2, MIN(pathavg)
      FROM dre_rte3d_yearpairs_candidates
      GROUP BY usentid, leafid1, leafid2)
;

-- Query sets using year pair tables.
--    -- ymd_to_ymd           August 1, 2003 to Sept 2, 2004


--          -- none in dev set. Some potentially useful queries:

--                -- dre_rte3d_yearpairs_terms
                  CREATE TABLE dre_rte3d_yearpairs_terms AS
                  SELECT usentid, termid, term
                  FROM rte3d_nwd
                  WHERE (usentid, termid) IN
                        (SELECT usentid, leafid1 FROM
dre_rte3d_yearpairs
                        UNION
                        SELECT usentid, leafid2 FROM
dre_rte3d_yearpairs)
                  ORDER BY usentid, termid
                  ;

--                -- Find rows with months.
--                SELECT a.usentid,
--                  a.termid AS yearid,
--                  b.termid AS monthid,
--                  a.term AS yearterm,
--                  b.term AS monthterm,
--                  a.termid - b.termid AS mydiff
--                FROM dre_rte3d_yearpairs_terms a JOIN rte3d_nwd b ON
a.usentid = b.usentid
--                WHERE b.term IN (SELECT monthterm FROM dre_months)
--                ;

--    -- ym_to_ym             August 2003 to Sept 2004

--          -- None in dev set.

--    -- y_to_y               2003 to 2004

      INSERT INTO dre_rte3d_final
      SELECT usentid,
      NULL AS nounid,
      NULL AS noun,
      rootid AS verbid,
      rootterm AS verb,
      1 AS startmonth,
      1 AS startday,
      leafterm1 AS startyear,
```

```
      leafid1 AS startyearid,
      12 AS endmonth,
      31 AS endday,
      leafterm2 AS endyear,
      leafid2 AS endyearid,
      'y_to_y' AS datetype
      FROM dre_rte3d_yearpairs
      ;

-- Find month pairs in sentences.
-- dre_rte3d_monthpair_candidates : all branches containing two months

CREATE TABLE dre_rte3d_monthpair_candidates AS
SELECT a.usentid,
      a.rootid,
      a.rootterm,
      a.leafid AS leafid1,
      b.leafid AS leafid2,
      a.leafterm AS leafterm1,
      b.leafterm AS leafterm2,
      a.termpath AS path1,
      b.termpath AS path2,
      (a.pathlength + b.pathlength)/2 AS pathavg,
      abs(a.pathlength - b.pathlength) AS pathdisc
FROM dre_rte3d_allpaths a JOIN dre_rte3d_allpaths b ON a.usentid =
b.usentid AND a.rootid = b.rootid
WHERE (a.leafterm) IN (SELECT monthterm FROM dre_months)
AND (b.leafterm) IN (SELECT monthterm FROM dre_months)
AND a.leafid < b.leafid
AND abs(a.pathlength - b.pathlength) < 3
AND (a.usentid, a.rootid) IN (SELECT usentid, verbid FROM
dre_rte3d_verbs)
AND (LOWER(a.termpath) LIKE '%from>%' OR LOWER(a.termpath) LIKE
'%between>%')
AND (LOWER(b.termpath) LIKE '%from>%' OR LOWER(b.termpath) LIKE
'%between>%' OR b.termpath LIKE '%to>%' OR b.termpath LIKE '%until>%')
ORDER BY a.usentid, a.rootid
;

-- dre_rte3d_monthpairs : sensible branches chosen from above ones.

CREATE TABLE dre_rte3d_monthpairs AS
SELECT * FROM dre_rte3d_monthpair_candidates
WHERE (usentid, leafid1, leafid2, pathavg) IN
      (SELECT usentid, leafid1, leafid2, MIN(pathavg)
      FROM dre_rte3d_monthpair_candidates
      GROUP BY usentid, leafid1, leafid2)
;

-- dre_rte3d_monthpairs_years : match the month pairs above with the
years.

CREATE TABLE dre_rte3d_monthpairs_years AS
SELECT a.usentid,
  b.rootid AS verbid,
  b.rootterm AS verb,
  leafid1,
```

```
  leafid2,
  leafterm1,
  leafterm2,
  a.termid AS yearid,
  a.term AS yearterm,
  a.termid - b.leafid2 AS mydiff
FROM dre_rte3d_candidates a JOIN dre_rte3d_monthpairs b ON a.usentid =
b.usentid
WHERE a.termid - b.leafid2 > 0 AND a.termid - b.leafid2 < 5
;

--    -- md-to-md_y           August 1 to Sept 2, 2003

--    -- dre_rte3d_monthpairs_mdy gets all the month, day, year info
with term ids et al.
      CREATE TABLE dre_rte3d_monthpairs_mdy AS
      SELECT c.*, d.termid AS numid2,
      d.term AS numterm2,
      d.numval AS numval2,
      c.numid1 - c.leafid1 AS diff1,
      d.termid - c.leafid2 AS diff2
      FROM
      (SELECT a.*, b.termid AS numid1, b.term AS numterm1, b.numval AS
numval1
      FROM dre_rte3d_monthpairs_years a JOIN dre_rte3d_numterms b ON
a.usentid = b.usentid) c JOIN dre_rte3d_numterms d ON c.usentid =
d.usentid
      WHERE c.numid1 < d.termid
      AND c.numval1 < d.numval
      AND c.numid1 - c.leafid1 < 3
      AND d.termid - c.leafid2 < 3
      ;

      INSERT INTO dre_rte3d_final
      SELECT c.usentid,
      NULL AS nounid,
      NULL AS noun,
      c.verbid,
      c.verb,
      c.monthnum1 AS startmonth,
      c.numval1 AS startday,
      c.yearterm AS startyear,
      c.yearid AS startyearid,
      d.monthnum AS endmonth,
      c.numval2 AS endday,
      c.yearterm AS endyear,
      c.yearid AS endyearid,
      'md-to-md_y' AS datetype
      FROM
      (SELECT a.*, b.monthnum AS monthnum1
      FROM dre_rte3d_monthpairs_mdy a
      JOIN dre_months b
      ON a.leafterm1 = b.monthterm) c
      JOIN dre_months d ON c.leafterm2 = d.monthterm
      WHERE (c.usentid, c.yearid) NOT IN (SELECT usentid, startyearid
FROM dre_rte3d_final)
```

```
      AND (c.usentid, c.yearid) NOT IN (SELECT usentid, endyearid FROM
dre_rte3d_final)
      ;

--    -- m-to-m_y             August to September 2003

--          -- None in dev set.

-- Match years to months.

CREATE TABLE dre_rte3d_monthyear AS
SELECT
  rte3d_nwd.usentid,
  rte3d_nwd.termid AS monthid,
  dre_rte3d_candidates.termid AS yearid,
  dre_rte3d_candidates.termid - rte3d_nwd.termid AS mydiff,
  rte3d_nwd.term AS monthterm,
  dre_rte3d_candidates.term AS yearterm
FROM rte3d_nwd
JOIN dre_rte3d_candidates
ON rte3d_nwd.usentid = dre_rte3d_candidates.usentid
WHERE rte3d_nwd.term IN
      (SELECT monthterm FROM dre_months)
AND abs(dre_rte3d_candidates.termid - rte3d_nwd.termid) <= 5
AND (dre_rte3d_candidates.usentid, dre_rte3d_candidates.termid) NOT IN
      (SELECT usentid, startyearid FROM dre_rte3d_final)
AND (dre_rte3d_candidates.usentid, dre_rte3d_candidates.termid) NOT IN
      (SELECT usentid, endyearid FROM dre_rte3d_final)
ORDER BY usentid
;

CREATE TABLE dre_rte3d_my_verb_candidates AS
SELECT a.usentid,
a.rootid,
a.rootterm,
a.leafid,
b.term AS leafterm,
a.pathlength
FROM dre_rte3d_allpaths a
JOIN rte3d_nwd b ON a.usentid = b.usentid AND a.leafid = b.termid
WHERE (a.usentid, a.rootid) IN (SELECT usentid, verbid FROM
dre_rte3d_verbs)
AND (a.usentid, a.leafid) IN (SELECT usentid, monthid FROM
dre_rte3d_monthyear)
ORDER BY usentid
;

CREATE TABLE dre_rte3d_my_verbs AS
SELECT a.usentid,
a.rootid AS verbid,
a.rootterm AS verb,
a.leafid AS monthid,
a.leafterm AS monthterm,
b.yearid,
b.yearterm,
MIN(b.mydiff) AS mydiff
FROM
```

```
(SELECT * FROM dre_rte3d_my_verb_candidates
WHERE (usentid, leafid, pathlength) IN
(SELECT usentid, leafid, MIN(pathlength)
FROM dre_rte3d_my_verb_candidates
GROUP BY usentid, leafid)) a
JOIN dre_rte3d_monthyear b ON a.usentid = b.usentid AND a.leafid =
b.monthid
GROUP BY a.usentid, a.rootid, a.rootterm, a.leafid, a.leafterm,
b.yearid, b.yearterm
;



--    -- m_d-to-d_y           August 1 to 2, 2003

--          -- none in dev set.

--    -- m_d_y                August 1, 2003

      INSERT INTO dre_rte3d_final
      SELECT e.usentid,
      NULL AS nounid,
      NULL AS noun,
      e.verbid,
      e.verb,
      f.monthnum AS startmonth,
      e.dayval AS startday,
      e.yearterm AS startyear,
      e.yearid AS startyearid,
      f.monthnum AS endmonth,
      e.dayval AS endday,
      e.yearterm AS endyear,
      e.yearid AS endyearid,
      'm_d_y' AS datetype
      FROM
        (SELECT c.usentid,
        c.verbid,
        c.verb,
        c.monthid,
        c.dayid,
        c.yearid,
        c.monthterm,
        d.numval AS dayval,
        c.yearterm FROM
          (SELECT a.*,
          b.termid AS dayid,
          b.term AS dayterm,
          b.termid - a.monthid AS dmdiff
          FROM dre_rte3d_my_verbs a JOIN rte3d_nwd b ON a.usentid =
b.usentid
          WHERE b.term IN (SELECT numterm FROM dre_numbers)
          AND b.termid - a.monthid < a.mydiff
          AND b.termid - a.monthid > -2) c
        JOIN dre_numbers d ON c.dayterm = d.numterm
        ORDER BY c.usentid) e
      JOIN dre_months f ON e.monthterm = f.monthterm
      ;
```

```
--    -- m_d-d_y              August 1-2, 2003

      INSERT INTO dre_rte3d_final
      SELECT e.usentid,
      e.nounid,
      e.noun,
      e.verbid,
      e.verb,
      e.monthnum AS startmonth,
      f.numval1 AS startday,
      e.yearterm AS startyear,
      e.yearid AS startyearid,
      e.monthnum AS endmonth,
      f.numval2 AS endday,
      e.yearterm AS endyear,
      e.yearid AS endyearid,
      'm_d-d_y' AS datetype
      FROM
      (SELECT c.usentid,
      NULL AS nounid,
      NULL AS noun,
      c.verbid,
      c.verb,
      d.monthnum,
      c.yearid,
      c.yearterm,
      c.numrange
      FROM
      (SELECT a.*,
      b.termid AS rangeid,
      b.term AS numrange,
      b.termid - a.monthid AS rangediff
      FROM dre_rte3d_my_verbs a JOIN rte3d_nwd b ON a.usentid =
b.usentid
      WHERE b.term IN (SELECT numrange FROM dre_numranges)
      AND b.termid - a.monthid < a.mydiff
      AND b.termid - a.monthid > -2) c
      JOIN dre_months d ON c.monthterm = d.monthterm) e
      JOIN dre_numranges f ON e.numrange = f.numrange
      WHERE (usentid, e.yearid) NOT IN
            (SELECT usentid, startyearid FROM dre_rte3d_final)
      AND (usentid, e.yearid) NOT IN
            (SELECT usentid, endyearid FROM dre_rte3d_final)
      ;

--    -- m_y                        August 2003

      INSERT INTO dre_rte3d_final
      SELECT a.usentid,
      NULL AS nounid,
      NULL AS noun,
      a.verbid,
      a.verb,
      b.monthnum AS startmonth,
      1 AS starday,
      a.yearterm AS startyear,
```

```
      a.yearid AS startyearid,
      b.monthnum AS endmonth,
      b.numdays AS endday,
      a.yearterm AS endyear,
      a.yearid AS endyearid,
      'm_y' AS datetype
      FROM dre_rte3d_my_verbs a JOIN dre_months b
      ON a.monthterm = b.monthterm
      WHERE (a.usentid, a.yearid) NOT IN
        (SELECT usentid, startyearid FROM dre_rte3d_final)
      AND (a.usentid, a.yearid) NOT IN
        (SELECT usentid, endyearid FROM dre_rte3d_final)
      ;

-- m-m_y               August-September 2003

CREATE TABLE dre_rte3d_myr_candidates AS
SELECT c.*, d.yearid, d.yearterm
FROM dre_rte3d_allpaths c JOIN
(SELECT a.usentid,
a.termid AS monthid,
a.term AS monthterm,
b.termid AS yearid,
b.term AS yearterm,
b.termid - a.termid AS mydiff
FROM rte3d_nwd a JOIN dre_rte3d_candidates b ON a.usentid = b.usentid
WHERE a.term IN (SELECT monthrange FROM dre_monthranges)
AND abs(b.termid - a.termid) <= 3) d
ON c.usentid = d.usentid AND c.leafterm = d.monthterm
WHERE (c.usentid, c.rootid) IN (SELECT usentid, verbid FROM
dre_rte3d_verbs)
;

INSERT INTO dre_rte3d_final
SELECT a.usentid,
NULL AS nounid,
NULL AS noun,
a.rootid AS verbid,
a.rootterm AS verb,
b.monthnum1 AS startmonth,
1 AS startday,
a.yearterm AS startyear,
a.yearid AS startyearid,
b.monthnum2 AS endmonth,
b.month2days AS endday,
a.yearterm AS endyear,
a.yearid AS endyearid,
'm-m_y' AS datetype
FROM
(SELECT usentid, rootid, rootterm, leafid, leafterm, pathlength,
yearid, yearterm
FROM dre_rte3d_myr_candidates
WHERE (usentid, leafid, pathlength)
IN
  (SELECT usentid, leafid, MIN(pathlength)
  FROM dre_rte3d_myr_candidates
  GROUP BY usentid, leafid)) a
```

```
JOIN dre_monthranges b ON a.leafterm = b.monthrange
;

-- y                    2003

CREATE TABLE dre_rte3d_year_candidates AS
SELECT * FROM dre_rte3d_allpaths
WHERE (usentid, leafid) IN
(SELECT usentid, termid FROM dre_rte3d_candidates
WHERE (usentid, termid) NOT IN (SELECT usentid, startyearid FROM
dre_rte3d_final)
AND (usentid, termid) NOT IN (SELECT usentid, endyearid FROM
dre_rte3d_final))
AND (usentid, rootid) IN (SELECT usentid, verbid FROM dre_rte3d_verbs)
;

INSERT INTO dre_rte3d_final
SELECT usentid,
NULL AS nounid,
NULL AS noun,
rootid AS verbid,
rootterm AS verb,
1 AS startmonth,
1 AS startday,
leafterm AS startyear,
leafid AS startyearid,
12 AS endmonth,
31 AS endday,
leafterm AS endyear,
leafid AS endyearid,
'y' AS datetype
FROM dre_rte3d_year_candidates
WHERE (usentid, leafid, pathlength) IN
  (SELECT usentid, leafid, MIN(pathlength)
  FROM dre_rte3d_year_candidates
  GROUP BY usentid, leafid)
;

CREATE TABLE dre_rte3d_final_clean AS
SELECT c.usentid,
c.nounid,
c.noun,
c.verbid,
c.verb,
c.startmonth,
c.startday,
c.startyear,
c.startyearid,
c.endmonth,
c.endday,
d.term AS endyear,
c.endyearid,
c.datetype
FROM
(SELECT
a.usentid,
a.nounid,
```

```
a.noun,
a.verbid,
a.verb,
a.startmonth,
a.startday,
b.term AS startyear,
a.startyearid,
a.endmonth,
a.endday,
a.endyear,
a.endyearid,
a.datetype
FROM dre_rte3d_final a
JOIN rte3d_nwd b
ON a.usentid = b.usentid
AND a.startyearid = b.termid) c
JOIN rte3d_nwd d ON c.usentid = d.usentid AND c.endyearid = d.termid
;

CREATE TABLE dre_rte3d_final_date AS
SELECT usentid,
nounid,
noun,
verbid,
verb,
TO_DATE(startmonth || '/' || startday || '/' || startyear,
'MM/DD/YYYY') AS startdate,
startyearid,
TO_DATE(endmonth || '/' || endday || '/' || endyear, 'MM/DD/YYYY') AS
enddate,
endyearid,
datetype
FROM dre_rte3d_final_clean
;

-- Logic tables and logic.

-- All E sentences

CREATE TABLE dre_rte3d_final_e AS
SELECT b.pairid, b.sentid, b.source AS senttype, a.*
FROM dre_rte3d_final_date a
JOIN rte3dtext b ON a.usentid = b.usentid
WHERE source = 'E'
ORDER BY b.pairid, b.sentid, b.source
;

-- All H sentences.

CREATE TABLE dre_rte3d_final_h AS
SELECT b.pairid, b.sentid, b.source AS senttype, a.*
FROM dre_rte3d_final_date a
JOIN rte3dtext b ON a.usentid = b.usentid
WHERE source = 'H'
ORDER BY b.pairid, b.sentid, b.source
;
```

```
-- Create table to hold negative results.

CREATE TABLE dre_rte3d_results
(pairid           number,
entailment  varchar2(10))
;

-- These are negative sentences according to our calculations.
-- If H contains a year reference, but no given sentence does, then no.

INSERT INTO dre_rte3d_results
SELECT pairid, 'NO' as entailment
FROM dre_rte3d_final_h
WHERE pairid IN
(SELECT DISTINCT pairid FROM rte3dtext WHERE pairid NOT IN
(SELECT pairid FROM
(SELECT a.*,
b.pairid,
b.source AS senttype
FROM dre_rte3d_candidates a
JOIN rte3dtext b ON a.usentid = b.usentid
WHERE b.source = 'E')))
;

-- Divide each group into a "range" or "exact" class based on date
range type.

CREATE TABLE dre_rte3d_final_h_class AS
SELECT a.*,
  CASE
    WHEN datetype = 'y' THEN 'exact'
    WHEN datetype = 'm_y' THEN 'exact'
    WHEN datetype = 'm_d_y' THEN 'exact'
    ELSE 'range'
  END dateclass
FROM dre_rte3d_final_h a
;

CREATE TABLE dre_rte3d_final_e_class AS
SELECT a.*,
  CASE
    WHEN datetype = 'y' THEN 'exact'
    WHEN datetype = 'm_y' THEN 'exact'
    WHEN datetype = 'm_d_y' THEN 'exact'
    ELSE 'range'
  END dateclass
FROM dre_rte3d_final_e a
;


-- -------------------- Date comparisons --------------------------

-- Rows where H contains E (exact) -- pair passes

CREATE TABLE dre_rte3d_h_contain_e_exact AS
SELECT h.pairid,
```

```
h.usentid AS h_usentid,
h.startdate AS h_startdate,
h.enddate AS h_enddate,
h.datetype AS h_datetype,
h.dateclass AS h_dateclass,
e.usentid AS e_usentid,
e.startdate AS e_startdate,
e.enddate AS e_enddate,
e.datetype AS e_datetype,
e.dateclass AS e_dateclass
FROM dre_rte3d_final_h_class h
JOIN dre_rte3d_final_e_class e ON h.pairid = e.pairid
WHERE h.dateclass = 'exact'
AND e.dateclass='exact'
AND
((h.startdate < e.startdate AND h.enddate > e.enddate)
OR
(h.startdate = e.startdate AND h.enddate > e.enddate)
OR
(h.startdate < e.startdate AND h.enddate = e.enddate))
;

-- Rows where H equals E (exact) -- pair passes

CREATE TABLE dre_rte3d_h_equal_e_exact AS
SELECT h.pairid,
h.usentid AS h_usentid,
h.startdate AS h_startdate,
h.enddate AS h_enddate,
h.datetype AS h_datetype,
h.dateclass AS h_dateclass,
e.usentid AS e_usentid,
e.startdate AS e_startdate,
e.enddate AS e_enddate,
e.datetype AS e_datetype,
e.dateclass AS e_dateclass
FROM dre_rte3d_final_h_class h
JOIN dre_rte3d_final_e_class e ON h.pairid = e.pairid
WHERE h.dateclass = 'exact'
AND e.dateclass='exact'
AND (h.startdate = e.startdate AND h.enddate = e.enddate)
;

-- Rows where H equals E (range) -- pair passes.

CREATE TABLE dre_rte3d_h_equal_e_range AS
SELECT h.pairid,
h.usentid AS h_usentid,
h.startdate AS h_startdate,
h.enddate AS h_enddate,
h.datetype AS h_datetype,
h.dateclass AS h_dateclass,
e.usentid AS e_usentid,
e.startdate AS e_startdate,
e.enddate AS e_enddate,
e.datetype AS e_datetype,
e.dateclass AS e_dateclass
```

```
FROM dre_rte3d_final_h_class h
JOIN dre_rte3d_final_e_class e ON h.pairid = e.pairid
WHERE h.dateclass = 'range'
AND e.dateclass='range'
AND (h.startdate = e.startdate AND h.enddate = e.enddate)
;

-- General (y, m_y, m-m_y) date in H contains a range in E
appropriately -- pair passes

CREATE TABLE dre_rte3d_exact_contains_range AS
SELECT h.pairid,
h.usentid AS h_usentid,
h.startdate AS h_startdate,
h.enddate AS h_enddate,
h.datetype AS h_datetype,
h.dateclass AS h_dateclass,
e.usentid AS e_usentid,
e.startdate AS e_startdate,
e.enddate AS e_enddate,
e.datetype AS e_datetype,
e.dateclass AS e_dateclass
FROM dre_rte3d_final_h_class h
JOIN dre_rte3d_final_e_class e ON h.pairid = e.pairid
WHERE
  ((h.datetype = 'y' AND (e.datetype='m_d-d_y' OR e.datetype='md-to-
md_y' OR e.datetype = 'm-m_y'))
  OR
  (h.datetype = 'm_y' AND (e.datetype = 'm_d-d_y' OR e.datetype = 'md-
to-md_y')))
AND (h.startdate <= e.startdate AND h.enddate >= e.enddate)
;

-- Monthrange contains month-day range, with matching months. Passes.
-- We require the matching months so that this doesn't get considered a
match:
-- H: "February-June 1996" E: "March 1-April 27 1996"
-- We do want this to match though:
-- H: "February-June 1996" E: "February 13-June 25 1996"

CREATE TABLE dre_rte3d_mrange_cont_mdrange AS
SELECT h.pairid,
h.usentid AS h_usentid,
h.startdate AS h_startdate,
h.enddate AS h_enddate,
h.datetype AS h_datetype,
h.dateclass AS h_dateclass,
e.usentid AS e_usentid,
e.startdate AS e_startdate,
e.enddate AS e_enddate,
e.datetype AS e_datetype,
e.dateclass AS e_dateclass
FROM dre_rte3d_final_h_class h
JOIN dre_rte3d_final_e_class e ON h.pairid = e.pairid
WHERE (h.datetype = 'm-m_y' AND e.datetype = 'md-to-md_y')
AND TO_CHAR(h.startdate, 'MM') = TO_CHAR(e.startdate, 'MM')
AND TO_CHAR(h.enddate, 'MM') = TO_CHAR(e.enddate, 'MM')
```

```
AND (h.startdate <= e.startdate AND h.enddate >= e.enddate)
;

-- H falls outside E. Fails if doesn't have a member is passed
categories

CREATE TABLE dre_rte3d_h_outside_e AS
SELECT h.pairid,
h.usentid AS h_usentid,
h.startdate AS h_startdate,
h.enddate AS h_enddate,
h.datetype AS h_datetype,
h.dateclass AS h_dateclass,
e.usentid AS e_usentid,
e.startdate AS e_startdate,
e.enddate AS e_enddate,
e.datetype AS e_datetype,
e.dateclass AS e_dateclass
FROM dre_rte3d_final_h_class h
JOIN dre_rte3d_final_e_class e ON h.pairid = e.pairid
WHERE (h.startdate > e.enddate OR h.enddate < e.startdate)
;

-- H partially overlaps E. Fails if not a member of a passed category.

CREATE TABLE dre_rte3d_h_partial_e AS
SELECT h.pairid,
h.usentid AS h_usentid,
h.startdate AS h_startdate,
h.enddate AS h_enddate,
h.datetype AS h_datetype,
h.dateclass AS h_dateclass,
e.usentid AS e_usentid,
e.startdate AS e_startdate,
e.enddate AS e_enddate,
e.datetype AS e_datetype,
e.dateclass AS e_dateclass
FROM dre_rte3d_final_h_class h
JOIN dre_rte3d_final_e_class e ON h.pairid = e.pairid
WHERE (h.startdate > e.startdate AND h.enddate > e.enddate)
OR (h.startdate < e.startdate AND h.enddate < e.enddate)
;

-- E contains H. Fails if not member of passing classes.

CREATE TABLE dre_rte3d_e_contains_h AS
SELECT h.pairid,
h.usentid AS h_usentid,
h.startdate AS h_startdate,
h.enddate AS h_enddate,
h.datetype AS h_datetype,
h.dateclass AS h_dateclass,
e.usentid AS e_usentid,
e.startdate AS e_startdate,
e.enddate AS e_enddate,
e.datetype AS e_datetype,
e.dateclass AS e_dateclass
```

```
FROM dre_rte3d_final_h_class h
JOIN dre_rte3d_final_e_class e ON h.pairid = e.pairid
WHERE (h.startdate > e.startdate AND h.enddate < e.enddate)
;

-- Insert elements from fail class that are not also in pass class.

INSERT INTO dre_rte3d_results
SELECT pairid, 'NO' AS entailment
FROM
  (SELECT pairid FROM dre_rte3d_h_outside_e
  UNION
  SELECT pairid FROM dre_rte3d_h_partial_e
  UNION
  SELECT pairid FROM dre_rte3d_e_contains_h)
WHERE pairid NOT IN
  (SELECT pairid FROM dre_rte3d_h_contain_e_exact
  UNION
  SELECT pairid FROM dre_rte3d_h_equal_e_exact
  UNION
  SELECT pairid FROM dre_rte3d_h_equal_e_range
  UNION
  SELECT pairid FROM dre_rte3d_exact_contains_range
  UNION
  SELECT pairid FROM dre_rte3d_mrange_cont_mdrange)
;
```

7 References.

Adams, R., Nicolae, G., Nicolae, C., & Harabagiu, S. (2007). Textual entailment through extended lexical overlap and lexico-semantic matching. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic, June 2007. 119-124.

Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I. (2006). The second PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

Blake, C. (2007). The role of sentence structure in recognizing textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic, June 2007. 101-106.

Bobrow, D. G., Condoravdi, C., Crouch, R., de Paiva, V., Karttunen, L., King, T. H., et al. (2007). Precision-focused textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic, June 2007. 16-21.

Brown, A.C., McCray, A.T., and Srinivasan, S. (2000). *The SPECIALIST lexicon.* Bethesda, MD: National Libraries of Medicine. Retrieved March 16, 2008 from http://lexsrv3.nlm.nih.gov/SPECIALIST/Projects/lexicon/current/release/LEX/DOCS/techrpt.pdf

Clark, P., Murray, W.R., Thompson, J., Harrison, P., Hobbs, J., & Fellbaum, C. (2007). On the role of lexical and world knowledge in RTE3. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Parahprasing*, Prague, Czech Republic, June 2007. 54-59.

Condoravdi, C., Crouch, D., de Paiva, V., Stolle, R., & Bobrow, D.G. (2003) Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, 9. 38-45.

Dagan, I., Glickman, O., & Magnini, B. (2005). The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, 2005.

Delmonte, R., Tonelli, S., Boniforti, M.A.P., Bristot, A. (2005). VENSES – a Linguistically-Based System for Semantic Evaluation. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, 2005.

Delmonte, R., Bristot, A., Boniforti, M.A.P., Tonelli, S. (2006). Coping with semantic uncertainty with VENSES. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

Dolan, B., Brockett, C., & Quirk, C.. (2005). *Microsoft Research paraphrase corpus*. Retrieved March 29, 2008 from http://research.microsoft.com/nlp/msr_paraphrase.htm.

Giampiccolo, D., Magnini, B., Dagan, I., & Dolan, B. (2007). The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic, June 2007. 1-9.

Hickl, A., & Bensley, J. (2007). A discourse commitment-based framework for recognizing textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic, June 2007. 171-176.

Hickl, A., Bensley, J., Williams, J., Roberts, K., Rink, B., & Shi, Y. (2006). Recognizing textual entailment with LCC's GROUNDHOG system. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

Jang, S.B., Baldwin, J., & Mani, I. (2004). Automatic TIMEX2 tagging of Korean news. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(1). 51-65.

Klein, D. & Manning, C.D. (2003a). Fast Exact Inference with a Factored Model for Natural Language Parsing. In *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, Cambridge, MA: MIT Press, pp. 3-10.

Klein, D. & Manning, C.D. (2003b). Accurate Unlexicalized Parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pp. 423-430.

MacCartney, B., & Manning, C.D. (2007). Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic, June 2007. 193-200.

The metaphysics of causation. (2007). In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (online ed.,). Stanford, CA: Metaphysics Research Lab, Center for Study of Language and Information, Stanford University. Retrieved March 16, 2008 from http://plato.stanford.edu/entries/causation-metaphysics/

Monz, C., & de Rijke, M. (2001) Light-weight entailment checking for computational semantics. In *Proceedings of the Third Workshop in Computational Semantics,* Siena, Italy. 59-72.

Nicholson, J., Stokes, N., & Baldwin, T. Detecting entailment using an extended implementation of the basic elements overlap metric. In *Proceedings of the second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

Perez, D., & Alfonseca, E. (2005). Application of the Bleu alogrithm for recognising textual entailments. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, 2005.

Raina, R., Haghighi, A., Cox, C., Finkel, J., Michels, J., Toutanova, K., MacCartney, B., de Marneffe, M., Manning, C.D., & Ng., A.Y. (2005). Robust Textual Inference using Diverse Knowledge Sources. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, 2005.

Roddick, J.F., & Spiliopoulou, M. (2002). A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering*, 14(4). 750-767.

Rodrigo, A., Penas, A., Herrera, J., & Verdejo, F. (2007). Experiments of UNED at the third Recognising Textual Entailment Challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic, June 2007. 89-94.

Schilder, F. (2004). Extracting meaning from temporal nouns and temporal prepositions. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(1). 33-50.

Tatu, M., Iles, B., Slavick, J., Novischi, A., & Moldovan, D. (2006). COGEX at the Second Recognizing Textual Entailment Challenge. In *Proceedings of the second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

Tatu, M., & Moldovan, D. (2007). COGEX at RTE3. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic, June 2007. 22-27.

Verhagen, M., Mani, I., Sauri, R., Knippen, R., Jang, S.B., Littman, J., Rumshisky, A., Phillips, J., & Pustejovsky, J. (2005). Automating temporal annotation with TARSQI. In *Proceedings of the ACL 2005*.

Wong, K, Xia, Y., Li, W., & Yuan, C. (2005). An overview of temporal information extraction. *International Journal of Computer Processing of Oriental Languages*, 18(2). 137-152.