Extraction of Temporal Information from Texts in Swedish

Anders Berglund, Richard Johansson, Pierre Nugues

LTH, Department of Computer Science, Lund University
Box 118
SE-221 00 Lund, Sweden
d98ab@efd.lth.se, {richard, pierre}@cs.lth.se

Abstract

This paper describes the implementation and evaluation of a generic component to extract temporal information from texts in Swedish. It proceeds in two steps. The first step extracts time expressions and events, and generates a feature vector for each element it identifies. Using the vectors, the second step determines the temporal relations, possibly none, between the extracted events and orders them in time. We used a machine learning approach to find the relations between events. To run the learning algorithm, we collected a corpus of road accident reports from newspapers websites that we manually annotated. It enabled us to train decision trees and to evaluate the performance of the algorithm.

1. Previous Work

The logic of event ordering and automatic extraction of such information has been a research topic for over 20 years. Allen (1984) pioneered the field by creating a formal classification of temporal relations. He identified 13 different relations between pairs of temporal intervals. If Allen's relations were to be applied to the text below, a graph such as the one in Figure 1 could be created.

> *Två personer dog$_{e1}$ när en bil körde$_{e2}$ av vägen och krockade$_{e3}$ med ett träd. Bilen {körde om}$_{e4}$ en annan bil när föraren {tappade kontroll}$_{e5}$ över den.*

> 'Two people died$_{e1}$ when a car drove$_{e2}$ off the road and crashed$_{e3}$ into a tree. The car {was overtaking}$_{e4}$ another car when the driver {lost control}$_{e5}$ of it.'

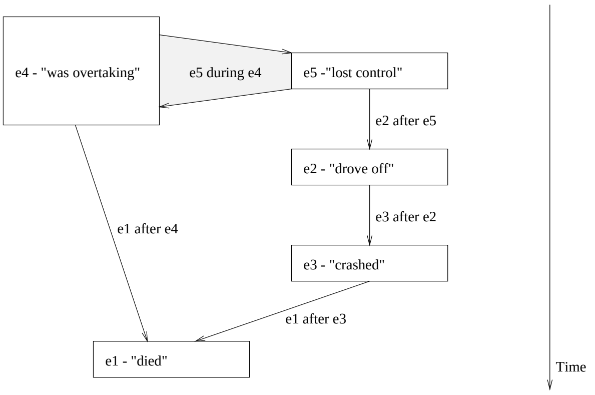

Figure 1: The chain of events in the example text.

Later, Dowty (1986) introduced the "narrative convention", the idea that the usage of two verbs in the perfect tense means that the second event occurs after the first one. In the accident report above, this implies that event $e3$ happens after event $e2$ as well as event $e5$ happening after event $e4$. It also implies that event $e4$ happens after event $e3$, which unfortunately is not true. Webber (1988) continued Dowty's work by creating a larger set of conventions for time stamping and ordering of phrases.

Lascarides and Asher (1993) presented a system that used a wealth of semantic knowledge to order events of phrases in pluperfect. Hitzeman et al. (1995) argued that such an approach is too complex, and work along those lines has been discontinued.

Machine learning techniques to extract time expressions and to determine temporal relations in texts in English are appearing. Verhagen et al. (2005), Boguraev and Ando (2005), and Mani and Schiffman (2005) are recent examples of them. Li et al. (2004) is another example for Chinese.

2. Temporal Information Processing

We designed and implemented a generic component to extract temporal information from texts in Swedish. The first step uses a pipeline of finite-state machines and phrase-structure rules that identifies time expressions and events. This step also generates a feature vector for each element it identifies. Using the vectors, the second step determines the temporal relations between the extracted events and orders them in time. In the rest of this article, we will focus on the second step, i.e., the detection of the relations between events.

We use a set of decision trees to find the relations between events. As input to the second step, the decision trees consider sequences of adjacent events, ranging from two to five, extracted by the first step and decide the temporal relation, possibly none, between pairs of them. We apply a transitive closure to these partial orderings to produce a temporal ordering for all the events in a text.

3. Corpus and Annotation

We automatically created the decision trees using the C4.5 machine learning program (Quinlan, 1992). As far as we know, there is no available time-annotated corpus in Swedish. We decided to collect and annotate a corpus of texts with temporal relations on which we trained the machine learning algorithm.

Several schemes have been proposed to annotate temporal information in texts. TimeML is an attempt to create a unified annotation standard for temporal information in texts (Pustejovsky et al., 2003a; Pustejovsky et al., 2005). Its goal is to capture most aspects of temporal relations between events in discourses. It is based on Allen's relations and a variation of Vendler's classification of verbs. It defines XML elements to annotate time expressions and events. Most notably, `TLINK`s describe the temporal relation holding between events or between an event and a time. TimeML is still an evolving standard (the latest annotation guidelines are from October 2005), and TimeBank (Pustejovsky et al., 2003b), the annotated corpus in English, is still rather small.

As development and test sets, we collected approximately 300 reports of road accidents from various Swedish newspapers. Each report is annotated with its publishing date. Analyzing the reports is complex because of their variability in style and length. Their size ranges from a couple of sentences to more than a page. The amount of details is overwhelming in some reports, while in others most of the information is implicit. The complexity of the accidents described ranges from simple accidents with only one vehicle to multiple collisions with several participating vehicles and complex movements.

We manually annotated a subset of this corpus consisting of 25 texts, 476 events, and 1,162 temporal links using a subset of the TimeML scheme. The annotation of the training set for the decision trees was done by a single annotator. When the relation was difficult to classify, we removed it from the training set.

Annotation is difficult for humans as well as for machines and human interannotator agreement is low. The complexity of the annotation scheme, and the fact that a large part of the information to annotate is implicit, accounts for this phenomenon. Additionally, the question of how to evaluate the performance is still not completely settled. When evaluating the temporal links, we used the method proposed by Setzer and Gaizauskas (2001), which measures precision/recall on the transitive closure of temporal links.

4. The Decision Trees

To order the events in time and create the temporal links, we use a set of decision trees. We apply each tree to sequences of events to decide the order between a pair of events in each sequence. If $e_1, ..., e_n$ are the events in the sequence they appear in the text, the trees correspond to the following functions:

$$\begin{array}{l} f_{dt1}(e_i, e_{i+1}) \Rightarrow t_{rel}(e_i, e_{i+1}) \\ f_{dt2}(e_i, e_{i+1}, e_{i+2}) \Rightarrow t_{rel}(e_i, e_{i+1}) \\ f_{dt3}(e_i, e_{i+1}, e_{i+2}) \Rightarrow t_{rel}(e_{i+1}, e_{i+2}) \\ f_{dt4}(e_i, e_{i+1}, e_{i+2}) \Rightarrow t_{rel}(e_i, e_{i+2}) \\ f_{dt5}(e_i, e_{i+1}, e_{i+2}, e_{i+3}) \Rightarrow t_{rel}(e_i, e_{i+3}) \end{array}$$

The possible output values are *simultaneous*, *after*, *before*, *is_included*, *includes*, and *none*. As a set of features, the decision trees use attributes of the considered events, temporal cue words or expressions between them, and other parameters such as the number of tokens separating the pair of events. The temporal cue words are called "signals" in TimeML.

We used five decision trees in total. The first tree, $dt1$, considers two adjacent[1] events and orders them. A second and a third tree ($dt2$ and $dt3$) order adjacent events considering features of the two events as well as features from the preceding and succeeding event, respectively. A fourth tree ($dt4$) orders two events separated by a third event, using features from all three events. The fifth tree ($dt5$) orders events separated by two other events, using features from all four events in question.

We never apply the decision trees across time expressions as we noted that the decision trees performed very poorly in these cases. As a consequence, $dt1$ can be applied more often than the others as it only requires two events in sequence instead of 3 or more. Our motivation for having trees that order events spaced further apart ($dt4$, $dt5$) is that the resulting ordering can be more fine-grained, and the motivation for having trees $dt2$ and $dt3$ is that they consider more context.

4.1. Features

The decision trees use the features of the involved events, as well as some measures we believe are useful such as an indication of what temporal signals were found between the events.

Instead of the TimeML class attribute, the decision trees use the morphological structure of the events. Both, the class attributes and morphological structures, contain similar data, but as the number of the different morphological structures is greater than the number of classes, the structure carries more information.

Below we present the features for the simplest tree, $dt1$:

- `mainEventTense`: `none`, `past`, `present`, `future`, `NOT_DETERMINED`.
- `mainEventAspect`: `progressive`, `perfective`, `perfective_progressive`, `none`, `NOT_DETERMINED`.
- `mainEventStructure`: `NOUN`, `VB_GR_COP_INF`, `VB_GR_COP_FIN`, `VB_GR_MOD_INF`, `VB_GR_MOD_FIN`, `VB_GR`, `VB_INF`, `VB_FIN`, `UNKNOWN`.
- `relatedEventTense`: (as `mainEventTense`)
- `relatedEventAspect`: (as `mainEvent-Aspect`)
- `relatedEventStructure`: (as `mainEvent-Structure`)
- `temporalSignalInbetween`: `none`, `before`, `after`, `later`, `when`, `continuing`, `several`.
- `tokenDistance`: `1`, `2 to 3`, `4 to 6`, `7 to 10`, `greater than 10`.
- `sentenceDistance`: `0`, `1`, `2`, `3`, `4`, `greater than 4`.

[1] Adjacent in the narrative order of the text.

- `punctuationSignDistance`: `0`, `1`, `2`, `3`, `4`, `5`, `greater than 5`.

The other trees use similar features, including the features of the other events involved in the query.

4.2. Applying the Trees

Figure 2 shows a part of C4.5's output for $dt1$. From this tree, we can extract the rule that when we consider a pair of adjacent events whose first one (`mainEvent`) is in the preterit tense and the second one (`relatedEvent`) is in the past perfect tense, the first event occurs after the second one in time. Figure 3 shows the application of this rule to the pair of simple sentences, *Bilen krockade med ett träd. Föraren hade druckit alkohol*, 'The car crashed against a tree. The driver had drunk alcohol'.

As Figure 2 shows, the C4.5 program also outputs pairs of numbers for each leaf of the decision trees. The first number is the "weight" of all queries reaching the leaf in question whereas the second one is the weight of the queries that were erroneously answered. These numbers do not correspond directly to the number of times the leaf is reached, but they are an indication of the accuracy of the leaf.

We use these numbers to compute a score for every leaf of the trees. The score for a leaf is computed as $weight_{correct}/weight_{total}$. The score for each generated `TLINK` is $score_{tree} * score_{answer\_leaf}$, where $score_{tree}$ is $1-\{$C4.5's error estimate for the final tree$\}$. If the leaf has a weight of 0.0, no queries reached that leaf in the training set. We then set the score to the arbitrarily chosen value of 0.2.

We use these scores when we resolve temporal loops as described in Section 4.4.

4.3. Training Set and Performance

Table 1 shows the final training set sizes, the final error rates for the trees as well as C4.5's error estimate for the final tree. The size of the training sets for the trees varies because of the number of matches made; $dt1$ is applied many more times than e.g. $dt5$. The reason that $dt2$ and $dt3$ have different training set sizes although they are applied exactly as many times is that we removed some relations from the training set.

Tree	Size	Errors$_{final}$	C4.5's error estimate
$dt1$	449	36.3%	44.2%
$dt2$	382	37.5%	46.1%
$dt3$	384	39.3%	46.0%
$dt4$	220	30.9%	47.5%
$dt5$	221	34.5%	46.2%

Table 1: Training set sizes and error rates for decision trees $dt1$–$dt5$.

The error rate presented in Table 1 is quite high. Our strategy relies on the redundancy of the trees and the assumption that the `TLINKs` with the higher scores are correct when they conflict with links with lower scores. The conflicting `TLINKs` with the lower scores are invalidated when we resolve temporal loops.

4.4. Resolving Temporal Loops

Figure 4 shows the 12 `TLINK`s that can be expected between a chain of four events. These `TLINK`s often conflict, and therefore there is a need to remove some of them.

Instead of removing `TLINKs`, we add `TLINKs` to an initially empty set if their inclusion wouldn't introduce temporal conflicts. We add the `TLINKs` with the highest scores first, thus "removing" the conflicting `TLINKs` with the lowest score.

5. Results

5.1. Two Example Runs

The texts R123 and R129 below are two examples of car accident reports from our corpus. The translation to English is done word-for-word as the order and indices of the tokens are important. Also note in text R129 that in (1) the preposition *i* 'in' is necessary in Swedish, but it is missing in both versions and clause (2) is ungrammatical. These mistakes were made by the journalist who wrote the original text. As a rule, we did not edit the texts in our corpus.

> En **trafikolycka**$_2$ **inträffade**$_3$ i snöovädret vid Fårö kyrka i går förmiddag. En bil **körde**$_{14}$ av vägen och **fortsatte**$_{18}$ in i ett träd varpå en person **klämdes**$_{26}$ fast. Räddningstjänsten och ambulans **kom**$_{32}$ på plats. Det **fanns**$_{37}$ under gårdagskvällen inga uppgifter på hur pass allvarliga personskadorna **var**$_{47}$.
>
> Text R123. *Gotlands Tidningar*, 04 January 2003.

> A **trafic.accident**$_2$ **occured**$_3$ in the.snow.bad.weather by Fårö church yesterday forenoon. A car **drove**$_{14}$ off the.road and **continued**$_{18}$ in into a tree after.which a person **was.jammed**$_{26}$ stuck. The.rescue.service and ambulance **came**$_{32}$ to the.site. There **were**$_{37}$ during yesterday.evening no reports regarding how serious the.person.injuries **were**$_{47}$.
>
> Text R123. English translation.

> Fyra personer **fördes**$_3$ till sjukhus efter en **bilolycka**$_8$ på riksväg 66 vid Erikslund i Västerås vid tiotiden på söndagsförmiddagen. Enligt polisen **har**$_{23}$ ingen av dem livshotande skador. Två personbilar och en lastbil var **inblandade**$_{36}$ (1) **olyckan**$_{37}$, som **inträffade**$_{40}$ på Riksväg under E18 (2). Vägen **stängdes**$_{47}$ av från olyckplatsen söderut men **öppnades**$_{53}$ igen efter ett par timmar.
>
> Text R129. *Expressen*, 29 December 2002.

> Four persons **were.taken**$_3$ to hospital after a **car.accident**$_8$ on national.highway 66 by Erikslund in Västerås at the.ten.time on Sunday.forenoon. According [to] the.police **have**$_{23}$ none of them life.threatening injuries.

```
mainEventTense = past:
|   relatedEventTense = present: before (42.0/10.4)
|   relatedEventTense = future: before (0.0)
|   relatedEventTense = past:
|   |   relatedEventAspect = progressive: before (145.0/73.7)
|   |   relatedEventAspect = perfective: after (7.0/6.1)
|   |   relatedEventAspect = none: before (21.0/5.9)
|   |   relatedEventAspect = perfective_progressive:
|   |   |   sentenceDistance = 0: simultaneous (6.0/2.3)
|   |   |   sentenceDistance = 1: before (2.0/1.8)
|   |   |   sentenceDistance = 2: simultaneous (0.0)
|   |   |   sentenceDistance = 3: simultaneous (0.0)
|   |   |   sentenceDistance = 4: simultaneous (0.0)
|   |   |   sentenceDistance = gt4: simultaneous (0.0)
mainEventTense = present:
|   relatedEventTense = none: after (16.0/4.8)
|   relatedEventTense = past: after (37.0/13.5)
|   relatedEventTense = present: simultaneous (56.0/20.0)
|   relatedEventTense = future: simultaneous (0.0)
```

Figure 2: Part of C4.5's output for $dt1$.

Text	*Bilen krockade med ett träd. Föraren hade druckit alkohol.*
	'The car crashed against a tree. The driver had drunk alcohol.'
Analysis	Main event (*krockade*): tense = past, aspect = progressive
	Related event (*hade druckit*): tense = past, aspect = perfective
Decision tree	mainEventTense = past =>
	relatedEventTense = past =>
	relatedEventAspect = perfective =>
	mainEvent **after** relatedEvent =>
	krockade **after** *hade druckit*
	'crashed' **after** 'had drunk'

Figure 3: Applying $dt1$ to a simple sentence

> Two person.cars and a truck were **involved**$_{36}$ (1) **the.accident**$_{37}$, which **occurred**$_{40}$ on national.highway under E18 (2). The.road **was.closed**$_{47}$ off from the.accident.site southwards but **was.opened**$_{53}$ again after a couple [of] hours.
>
> Text R129. English translation.

Figures 5 and 6 show the screenshots of the final event ordering. A line connecting two boxes means that the event in the upper box precedes the one in the lower box. In Figure 5, both @26:*klämdes* 'was jammed' and @32:*kom* 'came' are correctly ordered with respect to @14:*körde* 'drove' and @47:*var* 'were'. However, they are ordered incorrectly in respect to each other. In Figure 6, the event ordering is completely correct.

5.2. Interannotator Agreement

Interannotator agreement is known to be problematic in the context of temporal markup. In one pilot study, Setzer and Gaizauskas (2001), amongst other results, report a precision of 0.68 on average for the interannotator agreement for the classification of temporal relations. They used the same set of temporal relations that we used for our markup (i.e.,

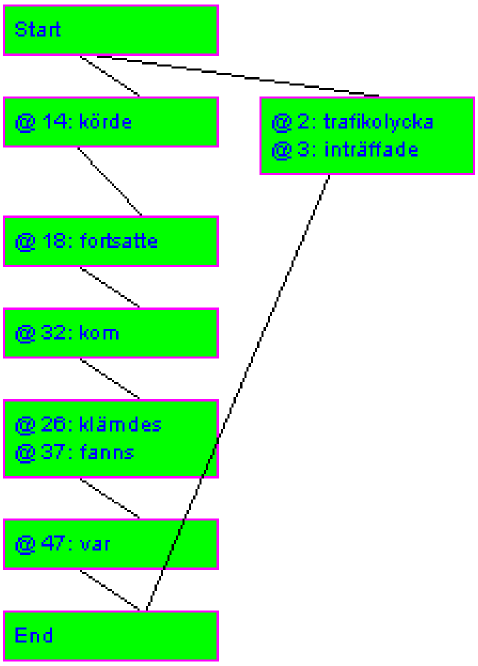

Figure 5: The event chain graph for text R123.

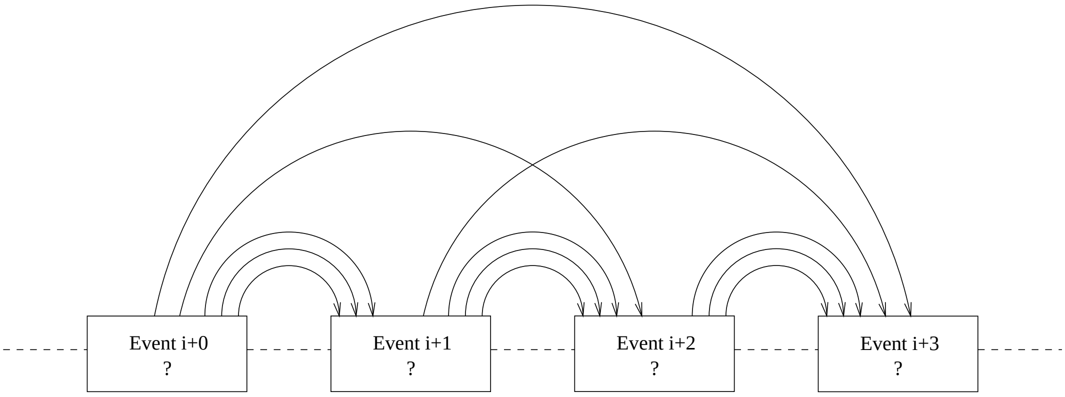

Figure 4: Between a sequence of four events, 12 `TLINKs` can be expected.

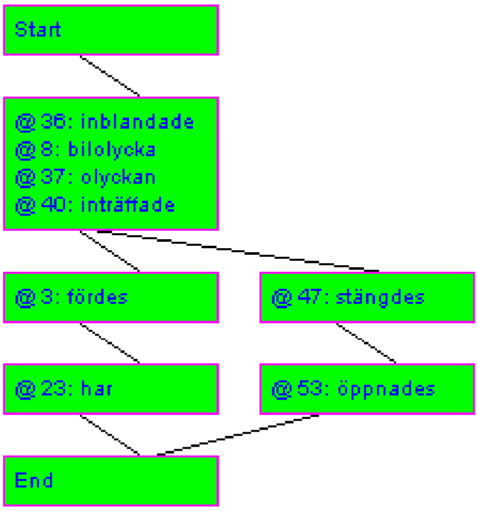

Figure 6: The event chain graph for text R129.

a subset of TimeML), and they also used newswire texts, so their measure of precision for interannotator agreement gives an indication of the difficulty of the problem.

6. Experimental Setup

We evaluated three aspects of the temporal information extraction: the detection of time expressions, the detection of events, and the quality of the final ordering. We considered that all the verbs and verb groups were events together with a small set of nouns. We built the trees automatically from this set using the C4.5 program.We report here the final ordering.

We applied a method proposed by Setzer and Gaizauskas (2001). They used the Cartesian product $(E \cup T) \times (E \cup T)$ where E denotes the set of all the events in the text and T, all the time expressions, and they denoted $S^{\vdash}$, $I^{\vdash}$, and $B^{\vdash}$, the transitive closures for the relations *simultaneous*, *includes*, and *before*, respectively.

If $S_k^{\vdash}$ and $S_r^{\vdash}$ represent the Gold Standard and the system response, respectively, for the set $S^{\vdash}$, the measures of precision and recall for the *simultaneous* relation are $R = \frac{|S_k^{\vdash} \cap S_r^{\vdash}|}{|S_k^{\vdash}|}$ and $P = \frac{|S_k^{\vdash} \cap S_r^{\vdash}|}{|S_r^{\vdash}|}$.

The overall measures of recall and precision are defined as: $R = \frac{|S_k^{\vdash} \cap S_r^{\vdash}| + |B_k^{\vdash} \cap B_r^{\vdash}| + |I_k^{\vdash} \cap I_r^{\vdash}|}{|S_k^{\vdash}| + |B_k^{\vdash}| + |I_k^{\vdash}|}$ and $P = \frac{|S_k^{\vdash} \cap S_r^{\vdash}| + |B_k^{\vdash} \cap B_r^{\vdash}| + |I_k^{\vdash} \cap I_r^{\vdash}|}{|S_r^{\vdash}| + |B_r^{\vdash}| + |I_r^{\vdash}|}$.

We limited our evaluation to the relations in the set $E \times E$ as our system doesn't support comparisons of time expressions..

7. Evaluation

We evaluated the temporal ordering created by the system for 10 previously unseen texts. We created a Gold Standard for these texts, and in order for us to judge their complexity relative to the texts used by Setzer and Gaizauskas, we also did an interannotator evaluation on the same texts where another member of our group also annotated the 10 texts.

Table 2 shows our results averaged over the 10 texts. As a reference, we also included Setzer and Gaizauskas averaged results for interannotator agreement on temporal relations in six texts in English. Note that Setzer and Gaizauskas did their evaluation over the set $(E \cup T) \times (E \cup T)$ instead of over $E \times E$.

Computing the transitive closure makes Setzer and Gaizauskas' evaluation method extremely sensitive. Missing a single link often results in a loss of scores of generated transitive links and thus has a massive impact on the final evaluation figures.

8. Application

We integrated this module, called TimeCore, in the Carsim program that generates 3D scenes from narratives describing road accidents (Johansson et al., 2005). TimeCore outputs its analysis in an XML format, and Carsim uses this information to order the events it detects. Many events are irrelevant for the visualization task and Carsim only uses a subset of the detected events. The temporal module enables the text-to-scene converter to animate the generated scene and visualize events described in the narrative.

9. Conclusion and Perspectives

We have developed a method for automatically detecting time expressions, events, and for ordering these events temporally. Although other systems have been described that extract temporal relations between pairs of events (Mani

Evaluation	Av. n_{words}	Av. n_{events}	P_{mean}	R_{mean}	F_{mean}
Gold vs. Automatic	98.5	14.3	54.85	37.72	43.97
Gold vs. Other Annotator	"	"	85.55	58.02	68.01
Setzer & Gaizauskas	312.2	26.7	67.72	40.07	49.13

Table 2: Evaluation results for final ordering with P, R, and F in %.

et al., 2003) or between clauses (Lapata and Lascarides, 2004), we believe we are the first to report results on the automatic ordering of events in complete narratives.

The work we have presented can be improved in several ways. The accuracy of the decision trees should improve with a larger training set. Switching from decision trees to other training methods such as Support Vector Machines could also improve results. The resolution of temporal loops could also gain from a global optimization instead of just discarding conflicting links.

10. References

James F. Allen. 1984. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154.

Branimir Boguraev and Rie Kubota Ando. 2005. TimeML-compliant text analysis for temporal reasoning. In *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 997–1003, Edinburgh, Scotland.

David R. Dowty. 1986. The effects of aspectual class on the temporal structure of discourse: Semantics or pragmatics? *Linguistics and Philosophy*, 9:37–61.

Janet Hitzeman, Marc Noels Moens, and Clare Grover. 1995. Algorithms for analyzing the temporal structure of discourse. In *Proceedings of the Annual Meeting of the European Chapter of the Association of Computational Linguistics*, pages 253–260, Dublin, Ireland.

Richard Johansson, Anders Berglund, Magnus Danielsson, and Pierre Nugues. 2005. Automatic text-to-scene conversion in the traffic accident domain. In *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 1073–1078, Edinburgh, Scotland, 30 July-5 August.

Mirella Lapata and Alex Lascarides. 2004. Inferring sentence-internal temporal relations. In *HLT-NAACL 2004: Main Proceedings*.

Alex Lascarides and Nicholas Asher. 1993. Temporal interpretation, discourse relations, and common sense entailment. *Linguistics & Philosophy*, 16(5):437–493.

Wenjie Li, Kam-Fai Wong, Guihong Cao, and Chunfa Yuan. 2004. Applying machine learning to Chinese temporal relation resolution. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 582–588, Barcelona.

Inderjeet Mani and Barry Schiffman. 2005. Temporally anchoring and ordering events in news (draft). In James Pustejovsky and Robert Gaizauskas, editors, *Time and Event Recognition in Natural Language*. John Benjamins.

Inderjeet Mani, Barry Schiffman, and Jianping Zhang. 2003. Inferring temporal ordering of events in news. In *Human Language Technology Conference (HLT'03)*, Edmonton, Canada.

James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003a. TimeML: Robust specification of event and temporal expressions in text. In *Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5)*, Tilburg, The Netherlands.

James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, and Marcia Lazo. 2003b. The TIMEBANK corpus. In *Proceedings of Corpus Linguistics 2003*, pages 647–656, Lancaster, United Kingdom.

James Pustejovsky, Robert Ingria, Roser Saurí, José Castaño, Jessica Littman, Robert Gaizauskas, Andrea Setzer, Graham Katz, and Inderjeet Mani. 2005. The specification language TimeML. In Inderjeet Mani, James Pustejovsky, and Robert Gaizauskas, editors, *The Language of Time: a Reader*. Oxford University Press.

J. Ross Quinlan. 1992. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Andrea Setzer and Robert Gaizauskas. 2001. A pilot study on annotating temporal relations in text. In *ACL 2001, Workshop on Temporal and Spatial Information Processing*, pages 73–80, Toulouse, France.

Marc Verhagen, Inderjeet Mani, Roser Saurí, Jessica Littman, Robert Knippen, Seok Bae Jang, Anna Rumshisky, John Phillips, and James Pustejovsky. 2005. Automating temporal annotation with TARSQI. In *Proceedings of the ACL 2005*.

Bonnie Webber. 1988. Tense as Discourse Anaphor. *Computational Linguistics*, 14(2):61–73.