

Temporal Signals Help Label Temporal Relations

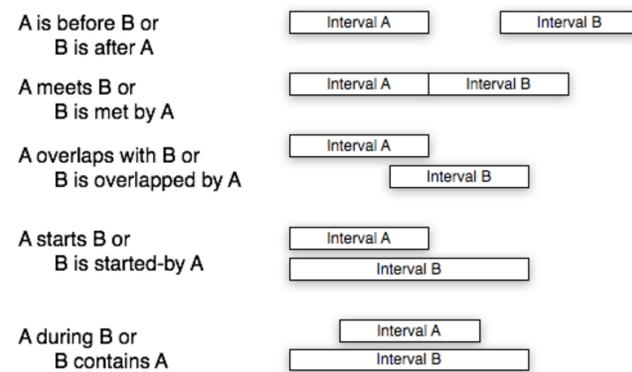
Leon Derczynski and Robert Gaizauskas

Problem: temporally ordering events and times

Mentions of events in text can often only be placed on a timeline by *relating* them to times and other events.

Relations describe the order of two intervals (events or times) w.r.t each other.

Allen (1983) describes 13 types, which look like:



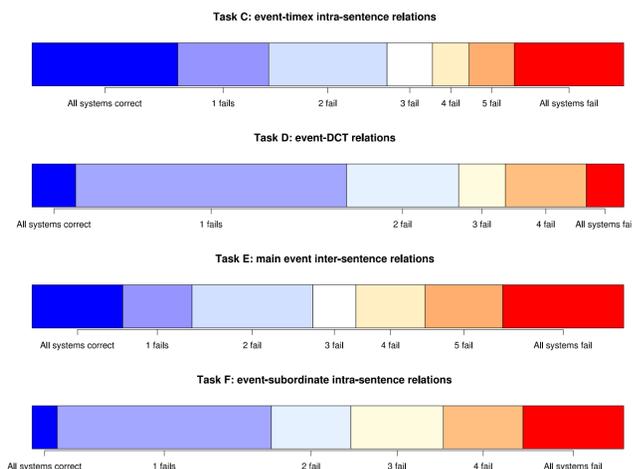
Relation *labeling* is the act of assigning one of these relation types to a relation between two intervals, thus temporally ordering them.

The state of the art is at an impasse

Many approaches and even evaluation exercises have tackled automatic general labeling of temporal relations (e.g. TempEval)

Accuracy rarely reaches above 60% for event-event links (the majority), or 80% for event-time links

Some links are difficult. What's in them?



About 30% of difficult links use a temporal signal

Hypothesis: explicit temporal signals can help ordering

Sometimes, a temporal relation is co-ordinated by a temporal *signal*

The torpedo was fired **after** the ship started sinking

This signal explicitly describes the nature of the temporal relation.

Signals may be a single word, as “after” above, or have a *head* and a *qualifier*.

We got out **just before** the storm hit

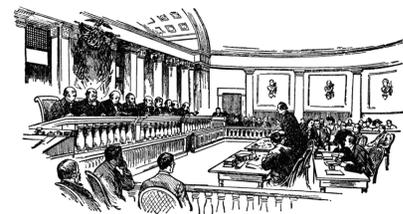
The match was won **shortly after**

Some signals are also polysemous:

I will drag you **before** the court!

This spatial use of “before” does not imply:

X ..and later, I will drag the court



Feature groups

Base features TimeML attributes event/time text, event class, part-of-speech, polarity, modality

In our corpus (TB-sig - adapted from TimeBank) 13.7% of relations had a co-ordinating signal.

Signal Text

The text of the signal is important; **before**, **during** and **after** all have different semantics
Signal, lowercase signal and signal head lemma are included

Argument Order

Two simple boolean features for argument interval position in text: ordering and same-sentence

Signal Order

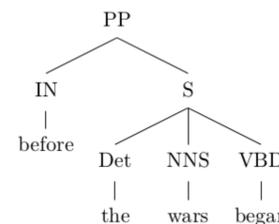
Relative position of the signal in the sentence affects temporal ordering:

You walk **before** you run
Before you walk you run

Syntax

We capture the constituent parse path between arguments, the label of the lowest common ancestor, and flags for interactions with a temporal function tag -TMP.

In this example, the path to began is IN-PP-S-VBD.



DCT

A flag for relations with times, indicating if the time is the document's default timestamp, which is often referred to implicitly.

Results

Yes, signals are a huge help: over 50% less error (+23% absolute performance)

Consistent improvements seen

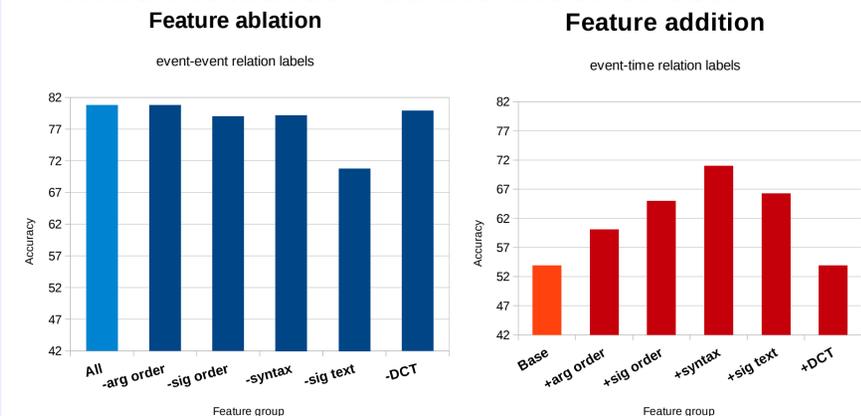
- When using this feature representation
- In many classifiers: MaxEnt, AdaBoost, Nbayes, LinearSVC, RandomForest

Features	Classifier	Event-event accuracy	Event-time accuracy
N/A	Baseline most-common-class	57.4%	51.6%
Base	Baseline maximum entropy	58.6%	59.6%
DG2010	Maximum entropy	72.6%	72.4%
	Random forest	76.7%	78.6%
	Adaptive boosting	70.4%	73.0%
	Naïve Bayes	73.8%	71.5%
All	Maximum entropy	75.5%	78.1%
	Linear SVC / Crammer-Singer	79.3%	75.6%
	Linear SVC	80.7%	77.1%
	Random forest	80.8%	80.3%

Interestingly, classification without signal features was much less effective on relations that used a signal than on those that didn't;
2.7% error reduction on signalled vs. **28%** on non-signalled (event-event)

Indicates one must take signal information into account to order these links.

Which features had the most effect?



Signal text

Intuitively, the signal word (**before**, **after**, **during** etc.) is very important
Removing these features gives a 10% accuracy drop – but, remaining features still give +14% absolute boost, even without the explicit ordering text

Syntax

When adding feature groups to the base set, group describing signal syntax gives the biggest boost (+13%) with event-time links (signal text best for event-event)

Honourable mention: order features

Adding just the Argument Order and Signal Order groups to the baseline gives almost all the performance of the full feature set – no use of text or syntax.