USFD2:Annotating Temporal Expressions and TLINKS for TempEval-2

Leon Derczynski (leon@dcs.shef.ac.uk) and Robert Gaizauskas (robertg@dcs.shef.ac.uk)

Task A: Annotating temporal expressions

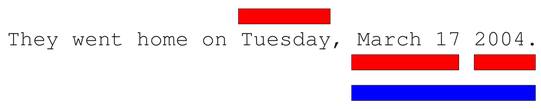
Determine the extent of the time expressions in a text as defined by the TimeML timex3 tag. In addition, determine value of the features type and val. The possible values of type are time, date, duration, and set; the value of val is a normalized value as defined by the timex2 and timex3 standards.

USFD2 uses rules to detect potential temporal expressions and then annotate them, including the anchoring of dates. This kind of approach generally increases in performance according to ruleset size.

1. Temporal expression boundary detection

We generate n-grams of input documents for *n* in 1 to 5, and then attempt to match input n-grams against a fixed ruleset. This is an extremely fast approach.

Initial matches are shown in red. These often overlap, in which case sub-matches are merged to have the widest boundary possible. A merged timex is shown in blue.



2. Temporal expression type classification

USFD2 only distinguishes between DATE and DURATION timexes.

DURATION rules:
- If the words "*for*" or "*during*" are included;
- If the timex ends with an "*s*";
- If there are only two tokens, and the first is "*a*".

DATE rules:
Anything else

Next week	► date
Four years	► duration
During that year	► duration
The next hours	► duration
March	► date

On the development data, this method achieves 90% accuracy.

3. Determine offset and direction, then anchor

We need to describe the size and temporal position of the expression in order to annotate it to TIMEX2 standard; further, DATEs should be normalised to a calendrical reference.

Shortcuts:
- Strings of "*today*" and "*now*" become `PRESENT_REF`
- Baldwin's 7-day sliding window provides accurate anchoring of independent week days
- Check if the string matches a range of known date formats

If these don't provide a reference:
- Detect number words - *"seven hundred"*
- Detect direction clues - "*next*", "*last*"
- Detect temporal units - "*days*", "*quarter*"

If there's no direction clue and no year, USFD2 attempts to anchor the day within *f* days after DCT, or failing that in the year prior to DCT. Setting *f*=14 works best with trial data.

Performance in Task A

Boundary detection:
P **0.84** R **0.79**
F1 **0.82**
TempEval2 worst and best F1: **0.26 - 0.86**

TIMEX type:
90% accuracy
TempEval2 worst and best F1: **84% - 98%**

TIMEX value:
17% accuracy
TempEval2 mean: **53.3%**
Accuracy on development data: **64%**

Tasks C & E: Labelling Temporal Relations

C. Determine the temporal relation between an event and a time expression in the same sentence. For TempEval-2, this task is further restricted by requiring that either the event syntactically dominates the time expression or the event and time expression occur in the same noun phrase.

E. Determine the temporal relation between two main events in consecutive sentences.

USFD2's temporal link classification is based on the work in Mani (2006), which used a set of features to describe two intervals and a maximum entropy classifier to make decisions. This is augmented with information about temporal signals, which has been shown to help.

1. Base feature set and classifier

For events:
- Tense (string)
- Aspect (string)
- Polarity (pos or neg)
- Modality (string)

For timexes:
- Type (timex type)
- Value (string)

For every interval:
- Token number in sentence / 5 (integer)
- The text for the interval (string)
- Interval type (event or timex)

For every relation:
- Arguments have same tense? (boolean)
- Arguments have same aspect? (boolean)

The MaxEnt classifier from NLTK is used.

2. Temporal signal annotation and meaning

Temporal signal words often describe the nature of a temporal link very clearly.

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

If we can provide features that describe this signal with regard to an appropriate temporal relation and two intervals that comprise it, then it is much easier to automatically label the temporal relation (Derczynski & Gaizauskas 2010).

Signals are not provided in the TempEval corpus. To annotated them, we used a list of TimeBank's signal texts, removing those entries with a low TFIDF. Every match in the TempEval data is considered a signal. This is a naïve annotation heuristic and its performance on the TempEval corpus is hard to evaluate.

Examples of signal text:
- *previous*
- *after*
- *throughout*

As we are only interesting in same- or successive-sentence temporal links, there are few instances where multiple signals are available. Where this happens, we prefer signals situated textually close to event texts and between temporal relation endpoints.

3. Signal features

We use signal hints to suggest a temporal relation type based on an Arg1-Signal-Arg2 text ordering:

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

The boolean features compensate for how temporal interpretation of signals may change depending on text order:

`**After** the torpedo was fired, the ship started sinking`

Signal features:
- Signal text (string)
- Signal hint (temporal relation type)
- Arg1 before signal? (boolean)
- Signal before arg2? (boolean)

The hints are compiled manually; here are some example mappings from text to relation type:

`thereafter`	► Before
`as of`	► Overlap-Or-After
`follows`	► After

Performance in Task C: Labelling relations between events and timexes in the same sentence

Accuracy: **63%**
TempEval2 worst and best: **0.54 - 0.65**

Performance in Task E: Labelling relations between head events in successive sentences

Accuracy: **45%**
TempEval2 worst and best: **0.45 – 0.58**

References:

Mani, Verhagen, Wellner, Lee, Pustejovsky. 2006. *Machine Learning of Temporal Relations*. In *Proceedings of the 21st International Conference on Computational Linguistics*.

Derczynski and Gaizauskas. 2010. *Using Signals to Improve Automatic Classification of Temporal Relations.* In *Proceedings of ESSLLI StuS, 2010*.