Evaluating TempEval Tasks

In full temporal annotation, evaluation of temporal annotation runs into the same issues as evaluation of anaphora chains: simple pairwise comparisons may not be the best way to evaluate. In temporal annotation, for example, one may wonder how the response in (1) should be evaluated given the key in (2). Scoring (1) at 0.33 precision misses the interdependce between the temporal relations. What we need to compare is not individual judgements but two partial orders.

(1) {A before B, A before C, B equals C}

(2) {A after B, A after C, B equals C}

For TempEval however, the tasks are defined in a such a way that a simple pairwise comparison is possible since we do not aim to create a full temporal graph and judgements are made in isolation.

Recall that there are three temporal relations: BEFORE, OVERLAP, and AFTER. In addition, we use three disjunctions over this set: BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER and VAGUE. The addition of these disjunctions raised the question on how to score a response of, for example, BEFORE given a key of BEFORE-OR-OVERLAP. We use two scoring schemes: strict and relaxed. The *strict scoring scheme* only counts exact matches as success. For example, if the key is OVERLAP and the response BEFORE-OR-OVERLAP than this is counted as failure. We can use standard definitions of precision and recall

$$Precision = R_c/R \qquad Recall = R_c/K$$

where Rc is number of correct answers in the response, R the total number of answers in the response, and K the total number of answers in the key. For the *relaxed scoring scheme*, precision and recall are defined as

$$Precision = R_cw/R \qquad Recall = R_cw/K$$

where R_cw reflects the weighted number of correct answers. A response is not counted as 1 (correct) or 0 (incorrect), but as one of the values in the following table.

	B	O	A	B-O	O-A	V
B	1	0	0	0.5	0	0.33
O	0	1	0	0.5	0.5	0.33
A	0	0	1	0	0.5	0.33
B-O	0.5	0.5	0	1	0.5	0.67
O-A	0	0.5	0.5	0.5	1	0.67
V	0.33	0.33	0.33	0.67	0.67	1

This scheme gives partial credit for disjunctions, but not so much that non-commitment edges out precise assignments. For example, assigning VAGUE as the relation type for every temporal relation results in a precision of 0.33.