Learning Word Relatedness over Time - Supplemental Material

1 Understanding Dynamics of Relations (Section 3.2)

Here we supply a detailed explanation and pseudo code for our peak detection algorithm.

Let $L = (t_1, v_1), (t_2, v_2), \ldots, (t_n.v_n)$ be a list of tuples, where t_i are time periods and v_i are values. Given L, Algorithm 1 returns a list of peak periods, i.e. $\{t \in L \mid t \text{ contains a peak}\}$.

The first step is to find relative maxima: we scan L and look for pairs (t_i, v_i) such that $v_{i-1} < v_i > v_{i+1}$, i.e. v_i is a local maximum. We apply two minimum thresholds in order to filter insignificant points: abs_{min} is an absolute threshold, which below it we consider the peak not to be significant enough. $coef_{min}$ is a relative threshold which facilitates removing points that are much lower than the highest maximum (lines 4–5 in the algorithm). In our experiments, we empirically chose over a validation set $abs_{min} = 0.1$, $coef_{min} = 0.8$. The final step of the algorithm is to find plateaus which will also be considered part of the peak. For each peak point, the algorithm considers the point's surrounding neighbors. Those identified as close in their value to the current peak, are added to the peak list.

2 Temporal Classifier (Section 3.3.2)

Algorithm 2 shows the pseudo code of the Temporal Classifier.

Algorithm 1: Peak detection algorithm

Input: $L = \{(t, v) \mid t$ are time periods, v are values$\}$, and constants $coef_{min}$, abs_{min}

Output: Peak list

```
1  begin
2      Peaks, RelativePeaks ← ∅
3      Maxima ← FindMaxima(L)
4      v_max ← max_v(L)
5      T_min ← max(v_max / coef_min, abs_min)
6      for (t, v) ∈ Maxima do
7          if v > T_min then
8              Add (t, v) to RelativePeaks
9          end
10     end
11     for peak = (t, v) ∈ RelativePeaks do
12         for (t_n, v_n) ∈ Neighbors(peak) do
13             if v_n is close to v then
14                 Add (t, v) to Peaks
15             end
16         end
17     end
18     return Peaks
19 end
```

Algorithm 2: Classifier using temporal model algorithm

Input: Temporal relation $r = (e_1, e_2, y)$ and classifier Cl_r
Output: True if e_1, e_2 are predicted to relate during year y, otherwise false

begin
  $History \leftarrow \emptyset$
  foreach *year* Y **do**
    $History[Y] \leftarrow cos(v_1^Y, v_2^Y)$
  end
  $Peaks \leftarrow PeakDetection(History)$
  $isPeak \leftarrow 1$ if $y \in Peaks$, o.w. 0
  $prediction \leftarrow Cl_r(v_1^y, v_2^y, y, isPeak)$
  return $true \iff prediction > 0$
end