

A Supplementary Material

Algorithm 1: Initial word-level annotation

Input: O : list with tokens of the original sentence, S : list with tokens of the simplified sentence, A : list with word alignments.

Output: SLO : simplification labels for each token in O , SLS : simplification labels for each token in S .

```

// labeling tokens in the original
// sentence
1 for  $i \leftarrow 1$  to  $\text{len}(O)$  do
    // get the indexes of the tokens
    // in  $S$  to which the  $i$ th token in
    //  $O$  is aligned to
2     $IS \leftarrow \text{FindAlignments}(A, i, 's')$ 
3    if  $\text{len}(IS) > 0$  then // it is aligned
4        if  $\text{len}(IS) = 1$  and  $O_i = S_{IS_0}$  then
5             $SLO_i \leftarrow 'C'$  // keep
6        else // not an exact match
7             $SLO_i \leftarrow 'R'$  // replace
8        end
9    else // not aligned
10        $SLO_i \leftarrow 'D'$  // delete
11    end
12 end
// labeling tokens in the simplified
// sentence
13 for  $j \leftarrow 1$  to  $\text{len}(S) + 1$  do
    // get the indexes of the tokens
    // in  $O$  to which the  $j$ th token in
    //  $S$  is aligned to
14     $IO \leftarrow \text{FindAlignments}(A, j, 'o')$ 
15    if  $\text{len}(IO) > 0$  then // it is aligned
16         $SLS_j \leftarrow 'O'$ 
17        if  $\text{len}(IO) > 1$  then
18            // the current token in  $S$ 
19            // replaces a phrase in  $O$ 
20            foreach  $k \in IO$  do
21                 $SLO_k \leftarrow 'R'$ 
22            end
23        else
24             $SLS_j \leftarrow 'A'$  // add
25    end
end

```

Algorithm 2: Annotation of reorderings

Input: SLO : simplification labels for each token in original sentence, SLS : simplification labels for each token in simplified sentence, A : list with word alignments.

Output: SLO modified.

```

1 shift_left  $\leftarrow 0$ 
2 for  $i \leftarrow 0$  to  $\text{len}(SLO)$  do
3     if  $SLO_i \in ['D', 'R']$  then
4         shift_left  $\leftarrow$  shift_left + 1
5     else
6          $IS \leftarrow \text{FindAlignments}(A, i, 's')$ 
7         if  $\text{len}(IS) > 0$  then
8              $k \leftarrow IS_0$  // index of the
            // aligned token in the
            // simplified sentence
9         else
10             $k \leftarrow i$  // index of the token
            // in the original sentence
11        end
12        shift_right  $\leftarrow 0$ 
13        for  $j \leftarrow 0$  to  $k$  do
14            if  $SLS_j \in ['AC', 'RW']$  then
15                shift_right  $\leftarrow$  shift_right + 1
16            end
17        end
18        if  $i - \text{shift\_left} + \text{shift\_right} \neq k$  then
19            switch  $SLS_i$  do
20                case 'C' do  $SLS_i \leftarrow 'M'$ 
21                case 'R' do  $SLS_i \leftarrow 'RM'$ 
22                case 'RW' do  $SLS_i \leftarrow 'RWM'$ 
23            end
24        end
25    end
26 end

```
