A Wrapper Relational Feature Selection Method (pseudo code)

**Algorithm 1** Wrapper Relational Feature Selection Method

**Input:** predicate set $S$, radius $r$, distance $d$, number of predicate sets $ns$, number of ranked predicate sets to retain $k$, measure $m$

**Output:** importance for each element in set $S$

1. $predicateSets = \text{GeneratePossiblePredicateSets}(S)$
2. $sampledPredicateSets = \text{SamplePredicateSets}(predicateSets, ns)$
3. **for each** $predicateSet_i$ **in** $sampledPredicateSets$ **do**
   1. $G_i = \text{RelationalDBGraphicalization}(predicateSet_i)$
   2. $X_i = \text{ExplicitFeatureExtraction}(G_i, r, d)$
   3. $results[predicateSet_i] = \text{StatisticalLearner}(X_i, measure)$
4. **end for**
5. $topkPredicateSets = \text{CalculateTopKPredicateSets}(results, k)$
6. **for each** $topkPredicateSet_i$ **in** $topkPredicateSets$ **do**
   1. $X_{k_i} = \text{ExplicitFeatureExtraction}(G_{k_i}, r, d)$ \{ $G_{k_i}$ is the graphicalization of $topkPredicateSet_i$ computed before \}
   2. solve SVM optimization problem
   3. let $\theta^*$ be the solution
   4. $predicateImportance[topkPredicateSet_i] = \text{ComputeSPredicateImportance}(\theta^*)$
7. **end for**
8. $\text{ComputeOverallPredicateImportance}(predicateImportance)$
B  Cumulative and Individual Feature Removal

Figure 1 (left) shows the E/R-diagram for the hedge cue detection task, as discussed in Section 6.1. The right part of the figure shows the graphicalization of an example interpretation, where a dependency relation exists between the determiner *the* and the noun *variable*, i.e., the first is a noun modifier of the latter.

![E/R diagram](image)

Figure 1: Left: E/R diagram modeling the hedge cue detection task. Attributes are represented as oval nodes. Right: Graphicalization of an example interpretation. Attributes are written inside the entity or relation vertex.

As indicated in Section 6.2, in order to show the quality of the ranking, we also examined the ROC curves where we cumulatively and individually remove the predicates (Figure 2). For the former, we removed the predicates cumulatively from the background knowledge in their ranked order according to the respective methods. As can be seen, the removal of the first 4 high-level relation features has the biggest influence on the performance. The results after the removal of all 10 high-level relation features while there are only 10 features ranked can be explained by the fact that the predicate to indicate the word ID needs to remain present since it forms the primary key in the relational database representation of the domain (see Section 2).

To show that the effect on the performance is not only due to a reduced number of features, we also removed the features individually. The ROC curve indicates that the first 4 ranked predicates (CW, RightOf, LeftOf, and Next) reduce the performance, resulting in a lower score then when all predicates are used. Removal of the other predicates results in ROC curves that are above the all feature setting, which indicates that their presence has a negative influence on the performance. This ultimately is the case for DH, which generates a large number of low-level graph features, since every word in the sentence has a dependency head.
Figure 2: Cumulative (top left: wrapper approach, top right: embedded approach) and individual feature removal (bottom)