Appendix for Prune-and-Score: Learning for Greedy Coreference Resolution

Chao Ma, Janardhan Rao Doppa†, J. Walker Orr, Prashanth Mannem
Xiaoli Fern, Tom Dietterich and **Prasad Tadepalli**
School of Electrical Engineering and Computer Science, Oregon State University
{machao,orr,mannemp,xfern,tgd,tadepall}@eecs.oregonstate.edu
† School of Electrical Engineering and Computer Science, Washington State University
jana@eecs.wsu.edu

Appendix A. Pseudocode for Pruning and Scoring Function Learning

Algorithm 1 Pruning Function Learning

Input: $\mathcal{D}$ = Training data, (I, A, T) = Search space, b = Pruning parameter

Initialize the set of ranking examples $\mathcal{R} = \emptyset$
for each training example $(x, y^*) \in \mathcal{D}$ **do**
  $s \leftarrow I(x)$ *// initial state*
  while not $T(s)$ **do**
    Generate an example R_t to imitate this search step
    Aggregate training data: $\mathcal{R} = \mathcal{R} \cup R_t$
    $s \leftarrow$ Apply a^* on s
  end while
end for
$\mathcal{F}_{prune}$ = Rank-Learner($\mathcal{R}$)
return pruning function $\mathcal{F}_{prune}$

Appendix B. Proof for Expressiveness of the Prune-and-Score Approach

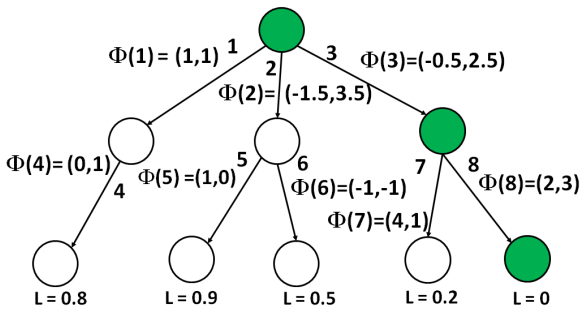

Figure 1: An example that illustrates that methods that use only scoring function for search can suffer arbitrary large loss compared to Prune-and-Score approach.

Proposition 1. *Let $\mathcal{F}_{prune}$ and $\mathcal{F}_{score}$ be functions from the same function space. Then for all learning problems, $\min_{\mathcal{F}_{score}} \mathcal{E}(\mathcal{F}_{score}, \mathcal{F}_{score}) \geq \min_{(\mathcal{F}_{prune}, \mathcal{F}_{score})} \mathcal{E}(\mathcal{F}_{prune}, \mathcal{F}_{score})$. Moreover there exist learning problems for which $\min_{\mathcal{F}_{score}} \mathcal{E}(\mathcal{F}_{score}, \mathcal{F}_{score})$ can be arbitrarily worse than $\min_{(\mathcal{F}_{prune}, \mathcal{F}_{score})} \mathcal{E}(\mathcal{F}_{prune}, \mathcal{F}_{score})$.*

Algorithm 2 Scoring Function Learning via Cross Validation

Input: $\mathcal{D}$ = Training data, $\mathcal{S}_p$ = Search space, b = Pruning parameter, $\mathcal{F}^*_{score}$ = Optimal scoring function

Divide the training set $\mathcal{D}$ into k folds $\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_k$
// Learn k different pruning functions
for $i = 1$ to k **do**
  $T_i = \cup_{j \neq i} \mathcal{D}_j$
  $\mathcal{F}^i_{prune}$ = **Learn-Pruner**$(T_i, \mathcal{S}_p, b)$
end for
// Generate ranking examples for scoring function training
Initialize the set of ranking examples $\mathcal{R} = \emptyset$
for $i = 1$ to k **do**
  for each training example $(x, y^*) \in \mathcal{D}_i$ **do**
    $s \leftarrow I(x)$ *// initial state*
    while not Terminal(s) **do**
      $A' \leftarrow$ Top b actions from $A(s)$ according to $\mathcal{F}^i_{prune}$
      $a^* \leftarrow \arg\max_{a \in A'} \mathcal{F}^*_{score}(s, a)$
      Generate ranking example R_t to imitate this search step
      Aggregate training data: $\mathcal{R} = \mathcal{R} \cup R_t$
      $s \leftarrow$ Apply a^* on s
    end while
  end for
end for
$\mathcal{F}_{score}$ = Rank-Learner($\mathcal{R}$)
return scoring function $\mathcal{F}_{score}$

Proof. The first part of the proposition follows from the fact that the first minimization is over a subset of the choices considered by the second. For the second part, consider a problem with a single training instance with search space shown in Figure 2. We assume linear $\mathcal{F}_{prune}$ and $\mathcal{F}_{score}$ functions of features $\Phi(n)$, where n is an action. The highlighted nodes correspond to the target path. The Prune-and-Score approach with $b = 2$ can find $\mathcal{F}_{prune}$ and $\mathcal{F}_{score}$ functions that are consistent with the target path. For example, with $\mathcal{F}_{prune} = (1, 0)$ and $\mathcal{F}_{score} = (1, 2)$ and pruning parameter 2 Prune-and-Score can achieve ze-

ro loss on this problem. However, it can be verified that there is no set of weights that satisfies all the constraints for imitating the target path by the Scoring-Only approach ($\mathcal{F}_{score}(3)>\mathcal{F}_{score}(2)$ and $\mathcal{F}_{score}(8)>\mathcal{F}_{score}(7)$ in particular). □