Prune-and-Score: Learning for Greedy Coreference Resolution

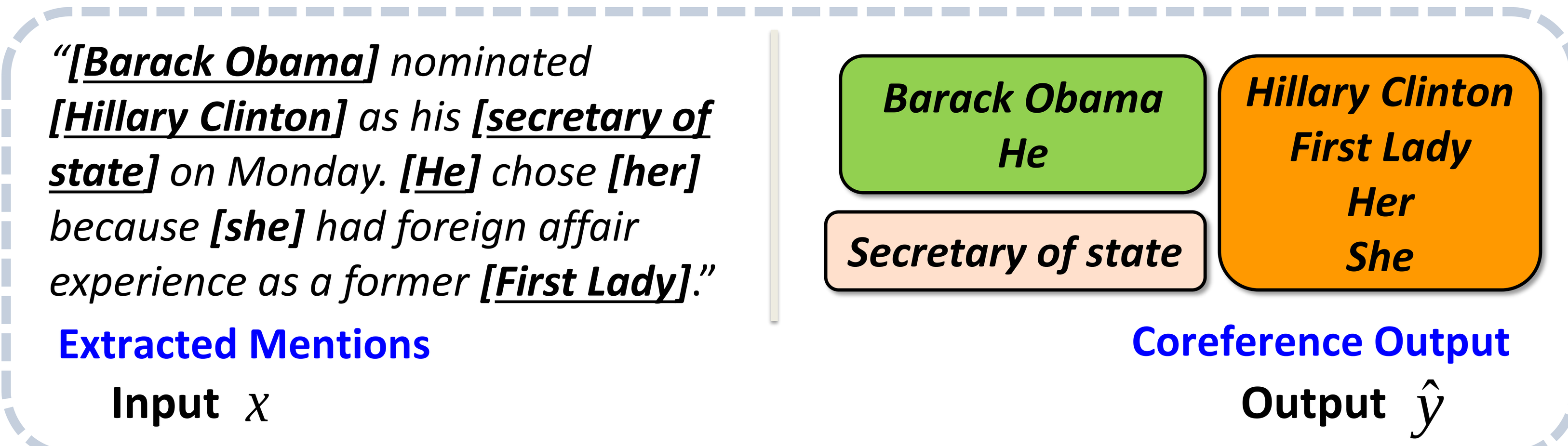
Chao Ma, Janardhan Rao Doppa, J. Walker Orr, Prashanth Mannem
Xiaoli Fern, Tom Dietterich and Prasad Tadepalli

Oregon State University

emnlp 2014
Doha, Qatar

Problem Setup

☐ **Coreference Resolution** is the task of clustering a set of mentions in the text such that all mentions in the same cluster refer to the same entity.



☐ **Learning:** Given a set of input-output pairs for training, learn a function $\mathcal{F}: \mathcal{X} \rightarrow \mathcal{Y}$ to make predictions on new inputs.

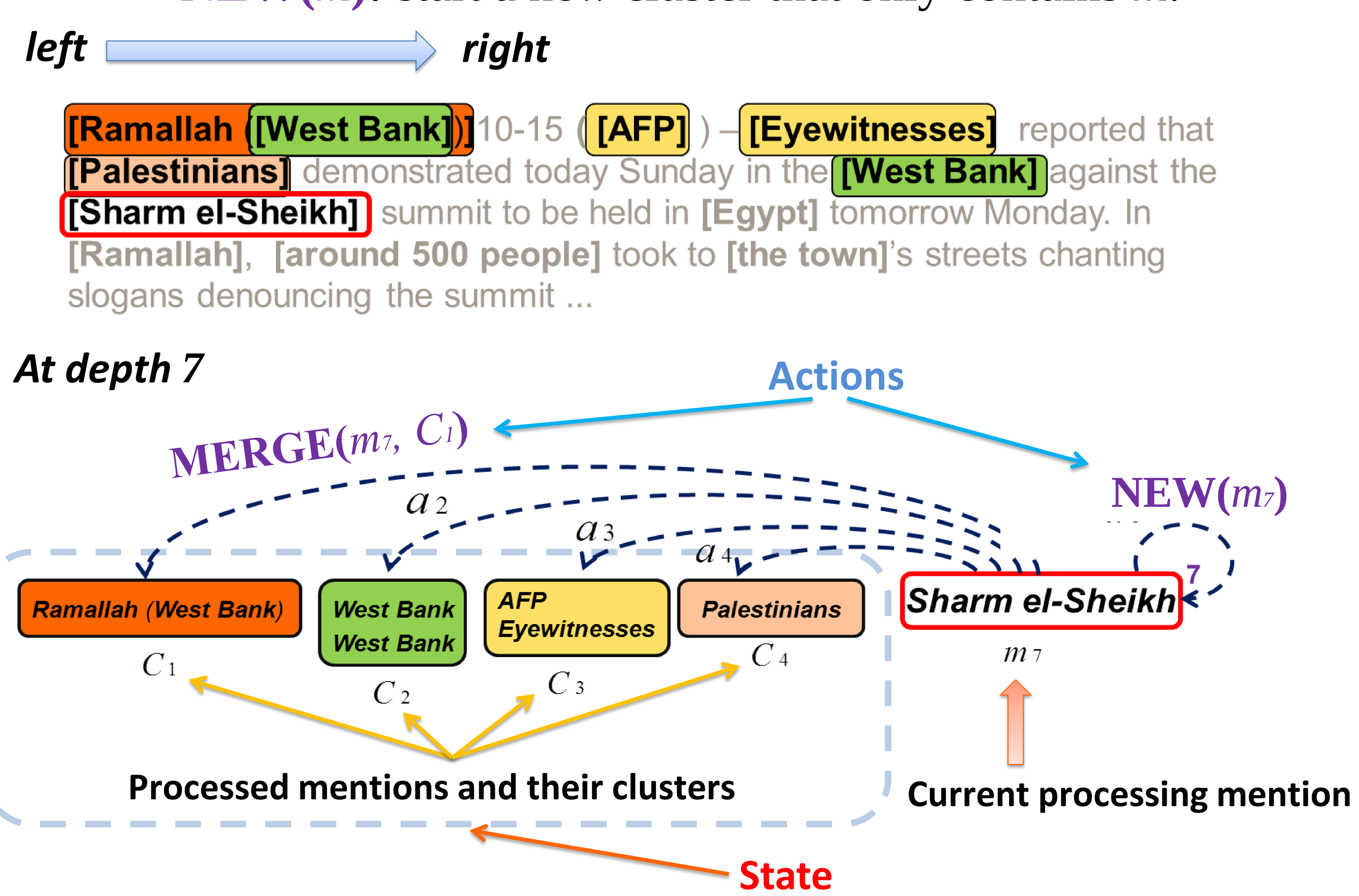
☐ **Evaluation:** against a non-negative loss $L(x, y, \hat{y}) \in R^+$ (e.g. BCubed).

Greedy Search Formulation

☐ **Greedy Search** processes each mention from left to right. Choose actions greedily according to a heuristic. "*Processed*" means an decision of that mention has been made.

☐ **Search Space**

- **State S:** Partial clustering of all mentions up to current mention.
- **Action:** MERGE(m, C): merge mention m into the cluster C.
  NEW(m): start a new cluster that only contains m.



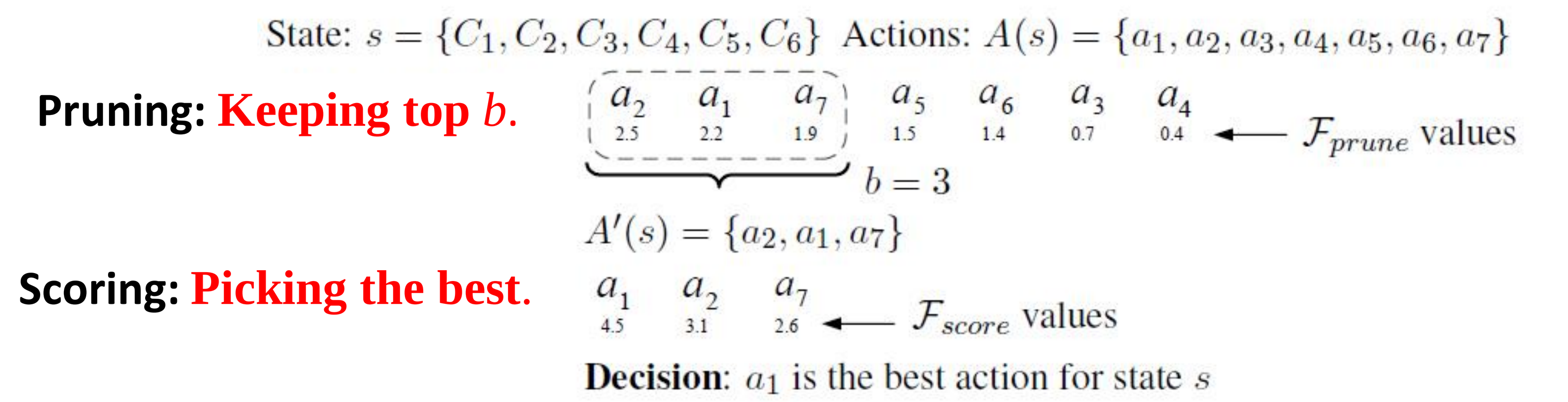
Each depth will have a corresponding processing mention;
The learned heuristic will pick the best action for that mention.

Prune & Score Framework

☐ **Key Idea:** Divide-and-conquer by learning two functions;

➤ A pruning function F_{prune} to prune all the bad decisions based on the specified pruning parameter b.

➤ A scoring function F_{score} to select the best decision from the remaining actions.



☐ **Representational Power**

Proposition: Let F_{prune} and F_{score} be in the same function space. For all learning problems, $\min_{F_{score}} \varepsilon(F_{score}, F_{score}) \geq \min_{(F_{prune}, F_{score})} \varepsilon(F_{prune}, F_{score})$. Moreover there exist learning problems for which $\min_{F_{score}} \varepsilon(F_{score}, F_{score})$ can be arbitrarily worse than $\min_{(F_{prune}, F_{score})} \varepsilon(F_{prune}, F_{score})$.

Loss Decomposition and Learning

☐ **Loss Decomposition**

Overall expected loss ε equals the error due to pruning the target output (ε_{prune}), plus the error due to not selecting the best output within the pruned space (ε_{score}).

$$\varepsilon = \varepsilon_{prune} + \varepsilon_{score|prune}$$

☐ **Pruning and Scoring Function Learning**

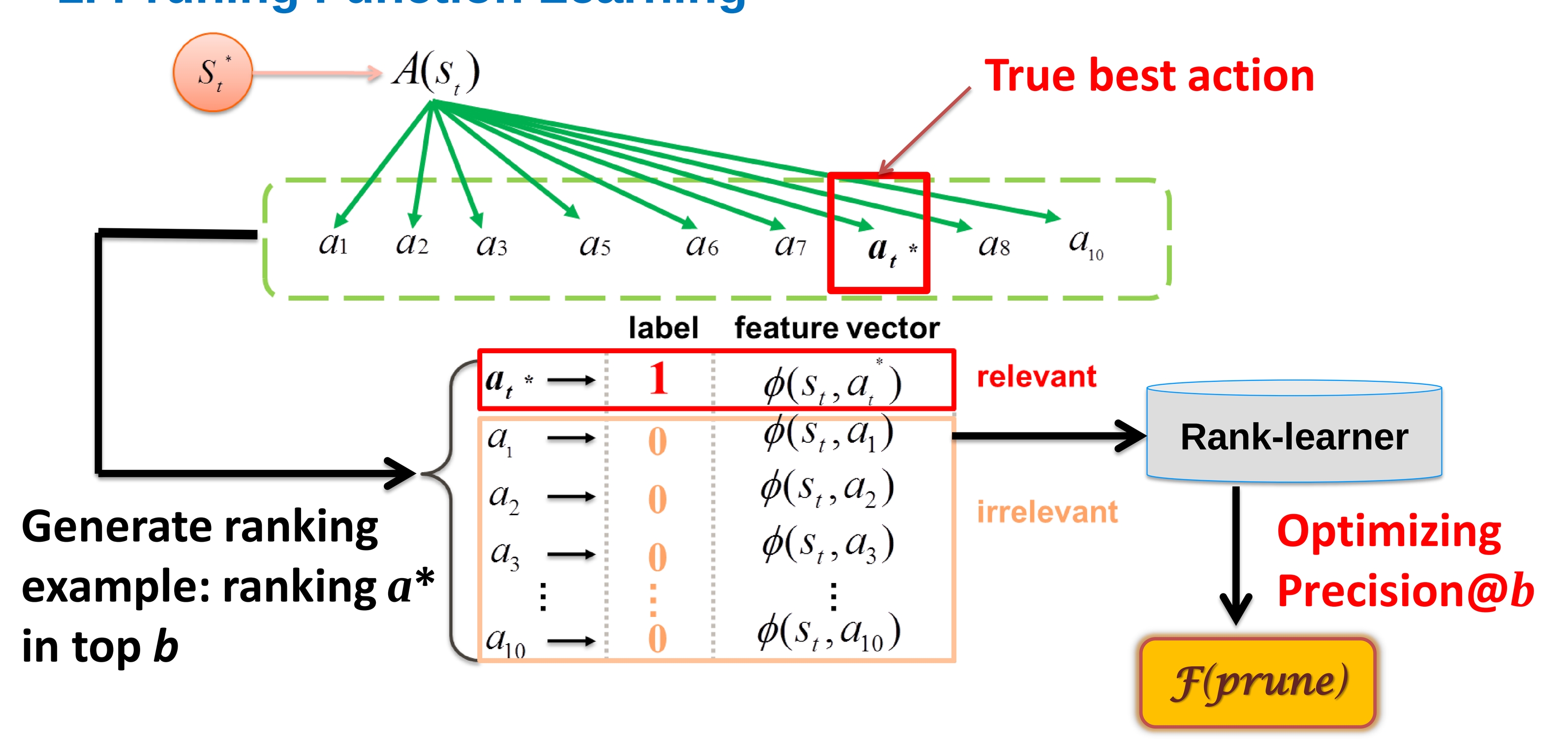
Stage 1: $\hat{\mathcal{F}}_{prune} \approx \arg\min_{\mathcal{F}_{prune} \in \mathbf{F_p}} \epsilon_{prune}$

Stage 2: $\hat{\mathcal{F}}_{score} \approx \arg\min_{\mathcal{F}_{score} \in \mathbf{F_s}} \epsilon_{score|\hat{\mathcal{F}}_{prune}}$

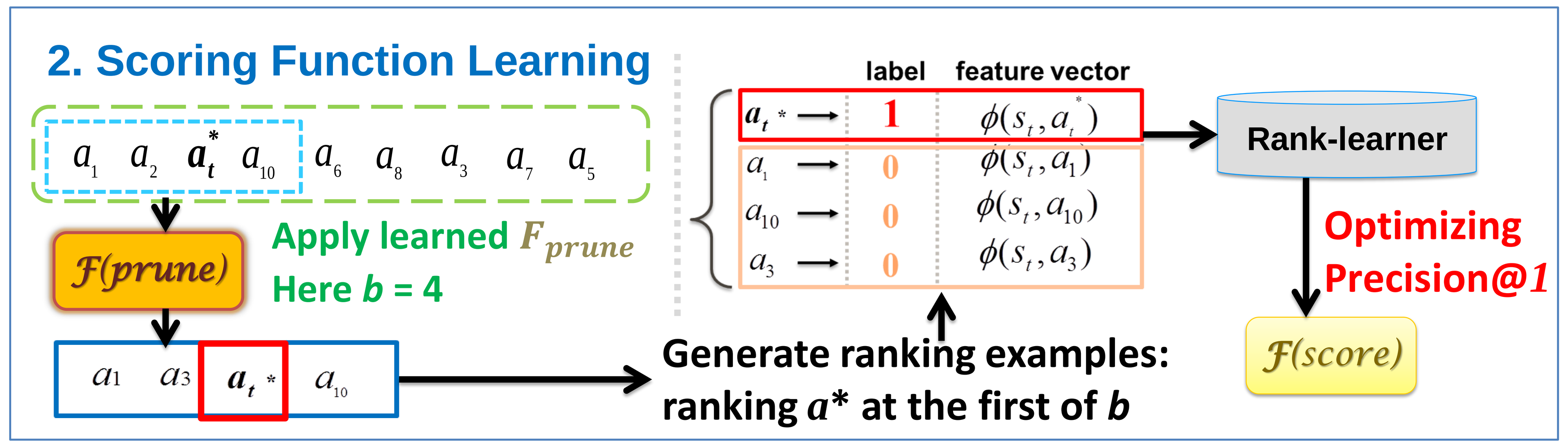
Conditioned on

☐ **Reductions to Rank-learning**

1. Pruning Function Learning



2. Scoring Function Learning



Experiment Results

☐ **Experiment Setups**

- **Datasets** **OntoNotes 5**: Train/Dev/Test: 2802/343/345 documents.
- **Base Rank-Learner** **LambdaMART** implemented in ***RankLib.***
- **Feature Set** Employ the same features as ***Easyfirst*** [Stoyanov et. al., 2012] System, which used **90 mention-pair** features; **49 entity-pair** features; and one **NEW indicator** feature.

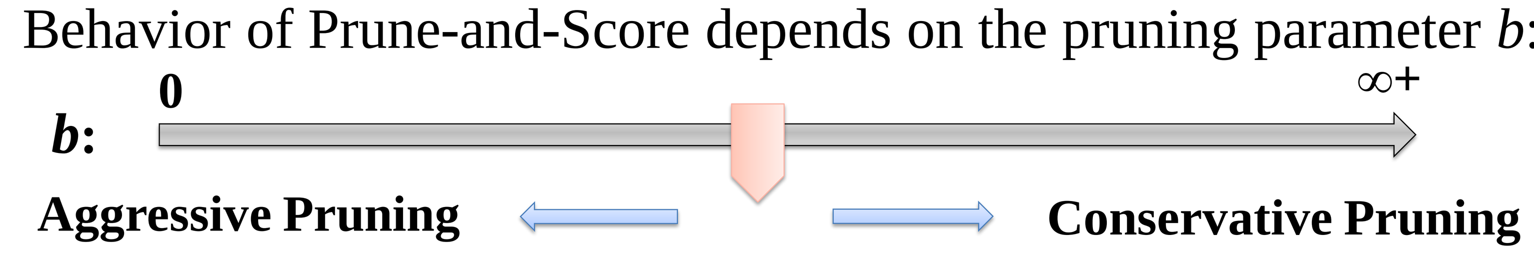
☐ **Coreference Resolution Results**

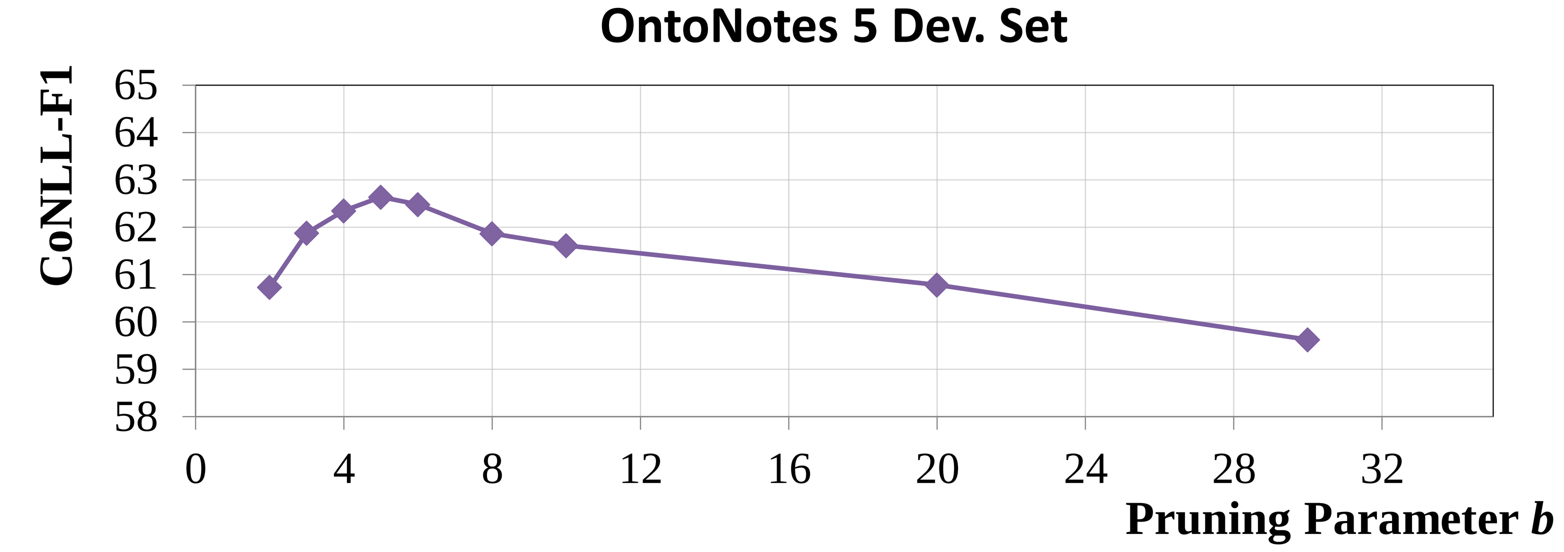
	OntoNotes 5.0 Test							
	System Mentions				Gold Mentions			
F-1 score	MUC	BCube	CEAF_e	CoNLL	MUC	BCube	CEAF_e	CoNLL
Prune-Score	**72.84**	57.94	53.91	61.56	86.96	76.49	**77.33**	**80.26**
Only Scoring	67.98	54.42	53.79	58.73	85.73	74.38	74.62	78.24
HOTCoref	70.72	**58.58**	**55.61**	**61.63**	-	-	-	-
Berkeley	70.82	58.14	55.27	61.41	**87.46**	76.63	76.40	80.16
UIUC	69.48	57.44	53.07	60.00	84.80	**78.74**	68.75	77.43
Stanford	64.71	52.26	49.32	55.43	83.64	74.81	66.98	75.14

- Prune-and-Score performs better than Only-Scoring. This shows the benefit of learning with pruning rules. Other coreference resolution systems can also benefit from our pruning idea.
- Prune-and-Score is comparable or better than the state-of-the-art.

☐ **Performance with Different Pruning Parameter b**

Behavior of Prune-and-Score depends on the pruning parameter b:





- Performance shows Prune-and-Score is robust to the pruning parameter b.