Deep Reinforcement Learning for Mention-Ranking Coreference Models

Kevin Clark and Christopher D. Manning
Stanford University

Presented by Zubin Pahuja

Coreference Resolution

- Identify all **mentions** that refer to the same real world entity

Barack Obama nominated ***Hillary Rodham Clinton*** as ***his secretary of state*** on ***Monday***. ***He*** chose ***her*** because ***she*** had ***foreign affairs*** experience as a former ***First Lady***.

Coreference Resolution

- Identify all **mentions** that refer to the same real world entity

Barack Obama nominated Hillary Rodham Clinton as ***his*** secretary of state on Monday. ***He*** chose her because she had foreign affairs experience as a former First Lady.

Coreference Resolution

- Identify all **mentions** that refer to the same real world entity
- A document-level structured prediction task

Barack Obama nominated Hillary Rodham Clinton as ***his*** secretary of state on Monday. ***He*** chose her because she had foreign affairs experience as a former First Lady.

Applications

- Full text understanding

  Information extraction, question answering, summarization

  "He was born in 1961"

Applications

- Dialog

  "Book tickets to see James Bond"

  "Spectre is playing near you at 2:00 and 3:00 today. How many tickets would you like?"

  "Two tickets for the showing at three"

Coreference Resolution is Hard!

- “She poured water from the pitcher into the cup until it was full”
- “She poured water from the pitcher into the cup until it was empty”

- The trophy would not fit in the suitcase because it was too big.
- The trophy would not fit in the suitcase because it was too small.

Coreference Resolution is Hard!

- “She poured water from the pitcher into the cup until it was full”
- “She poured water from the pitcher into the cup until it was empty”

- The trophy would not fit in the suitcase because it was too big.
- The trophy would not fit in the suitcase because it was too small.

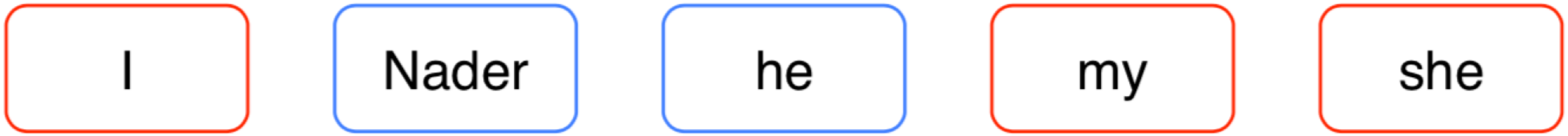
- These are called **Winograd Schema**

Three Kinds of Coreference Models

- Mention Pair
- Mention Ranking
- Clustering

Clustering

"***I*** voted for ***Nader*** because ***he*** was most aligned with ***my*** values," ***she*** said.

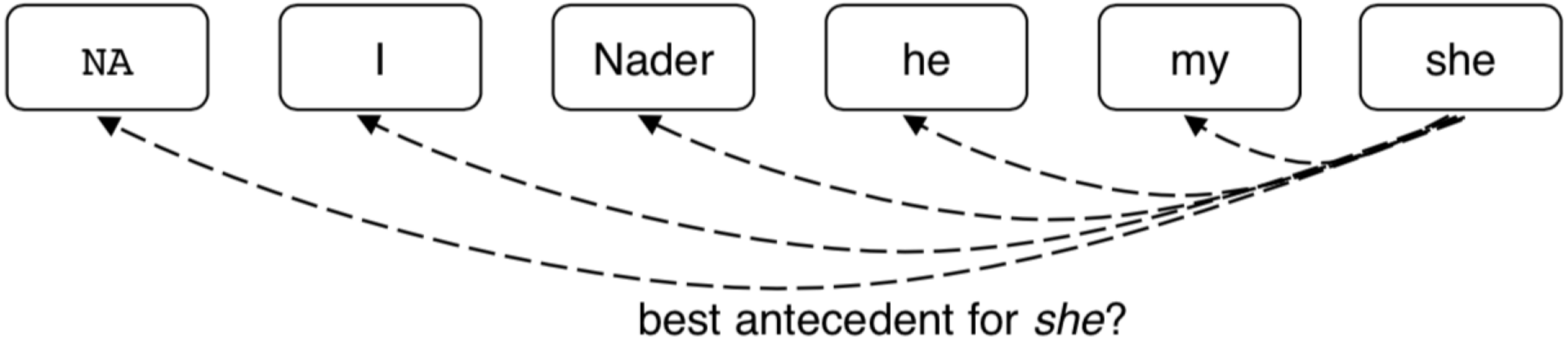


Coreference Cluster 1

Coreference Cluster 2

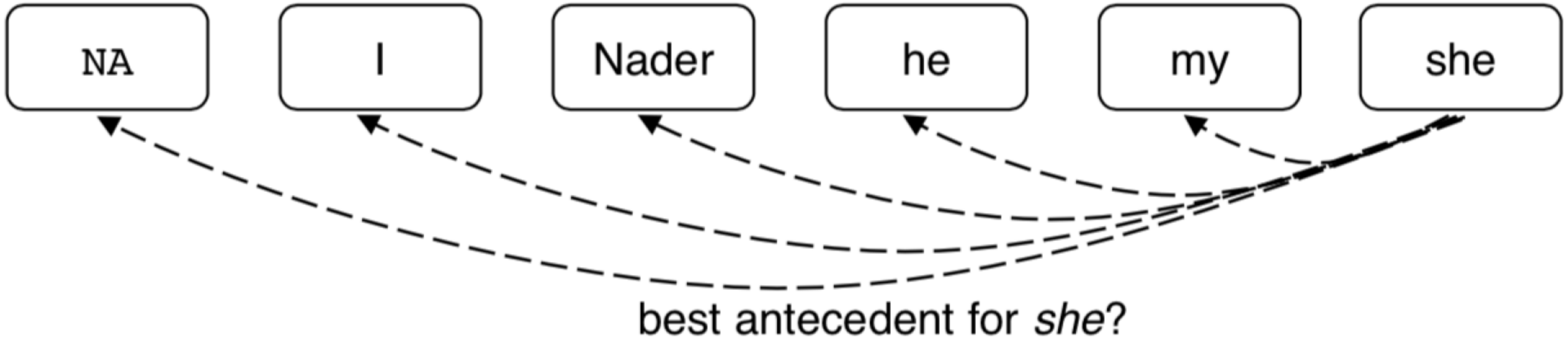
Mention Ranking

- Assign each mention its highest scoring candidate antecedent
- Dummy mention **NA** allows model to decline assigning antecedent to current mention



Mention Ranking

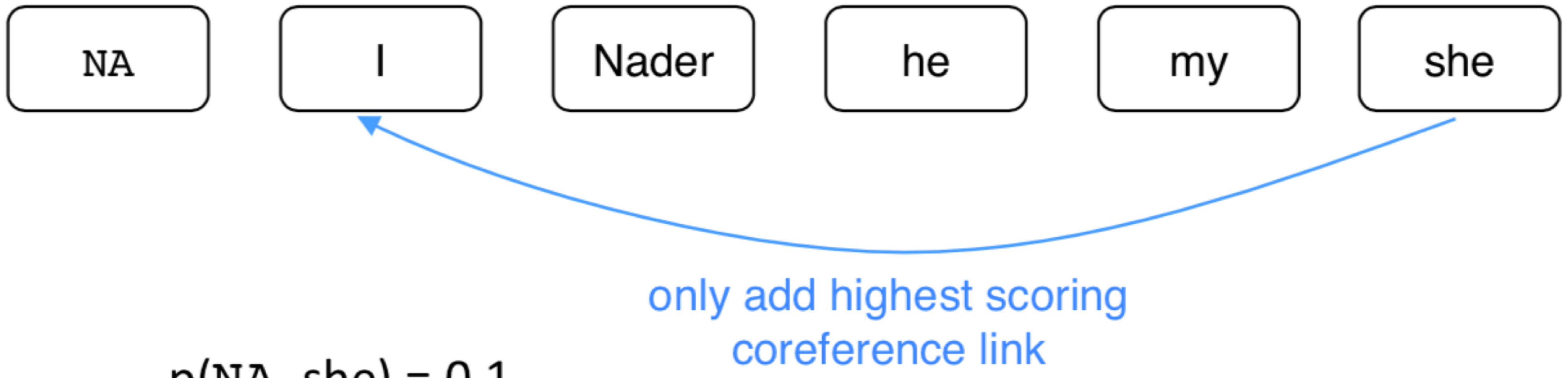
- Assign each mention its highest scoring candidate antecedent
- Dummy mention **NA** allows model to decline assigning antecedent to current mention



p(NA, she) = 0.1
p(I, she) = 0.5
p(Nader, she) = 0.1
p(he, she) = 0.1
p(my, she) = 0.2

Mention Ranking

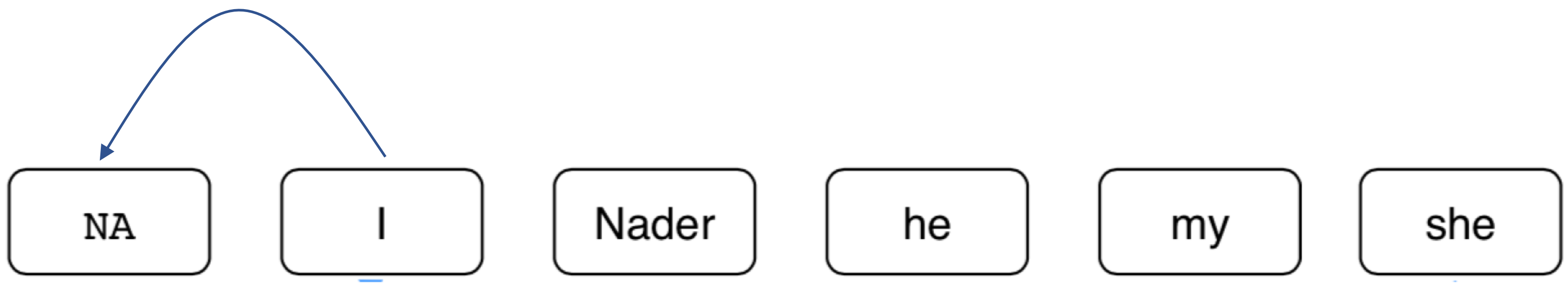
- Assign each mention its highest scoring candidate antecedent
- Dummy **NA** mention allows model to decline linking the current mention to anything



p(NA, she) = 0.1
p(I, she) = 0.5
p(Nader, she) = 0.1
p(he, she) = 0.1
p(my, she) = 0.2

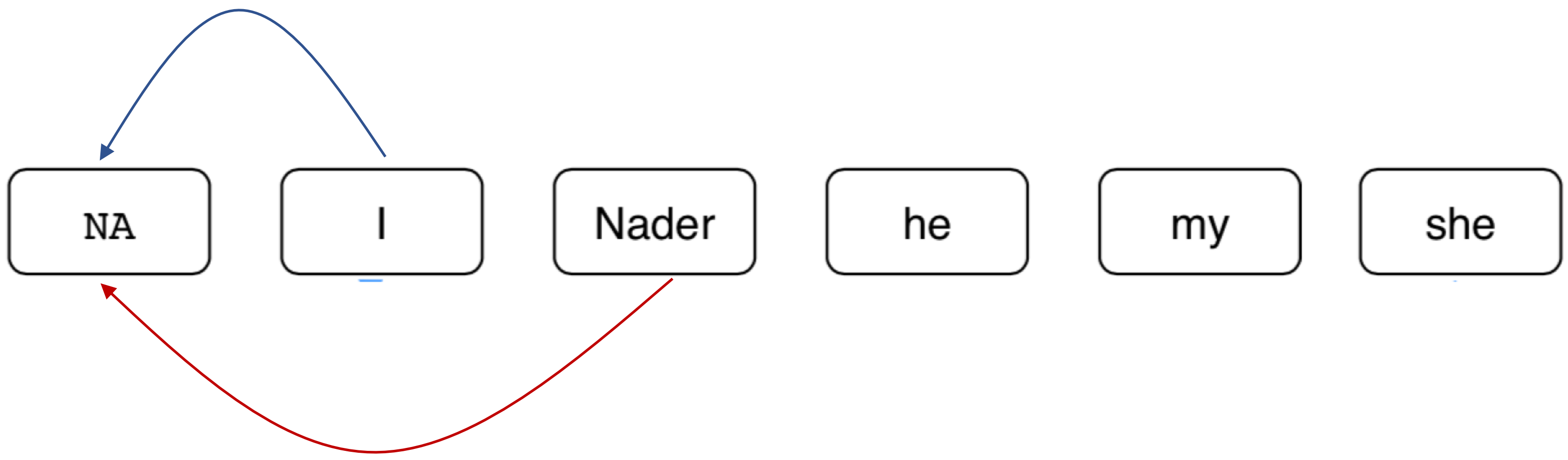
Mention Ranking

- Infer global structure by making a sequence of local decisions



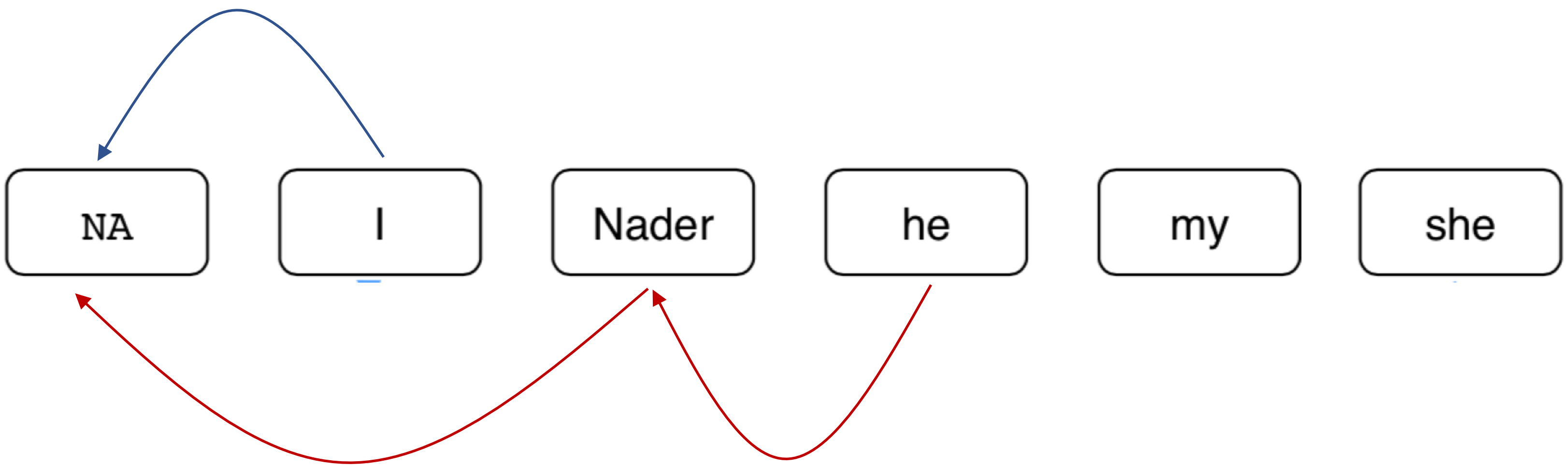
Mention Ranking

- Infer global structure by making a sequence of local decisions



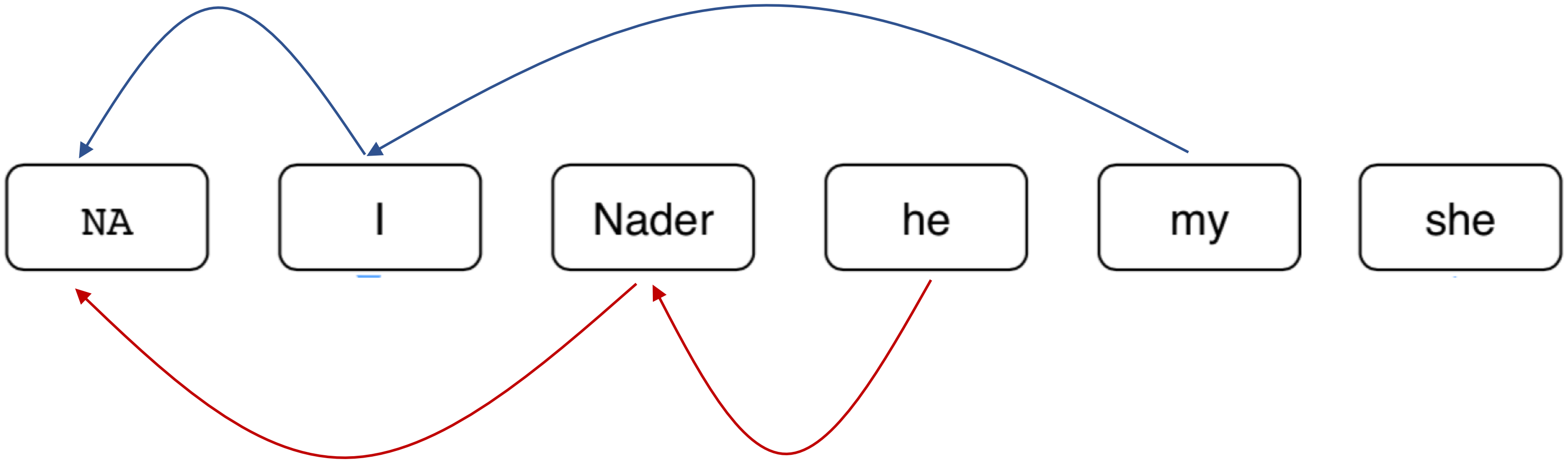
Mention Ranking

- Infer global structure by making a sequence of local decisions



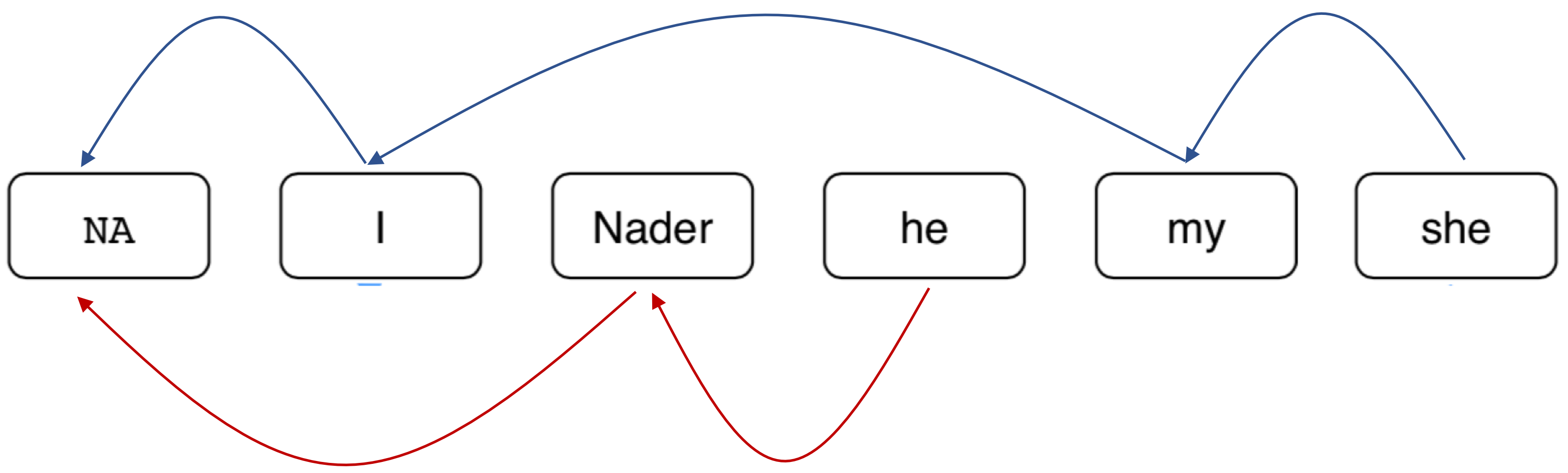
Mention Ranking

- Infer global structure by making a sequence of local decisions



Mention Ranking

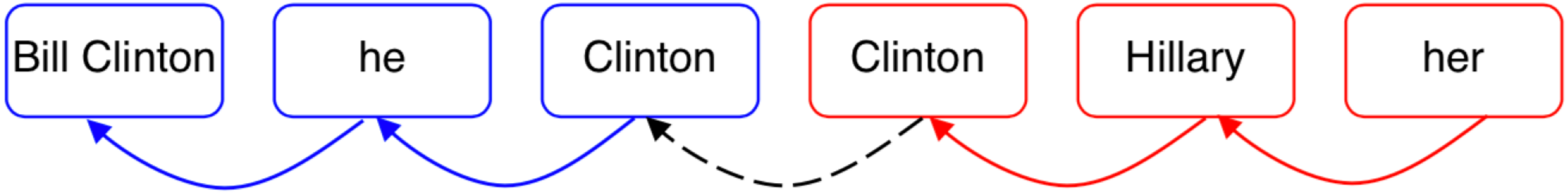
- Infer global structure by making a sequence of local decisions



Challenge

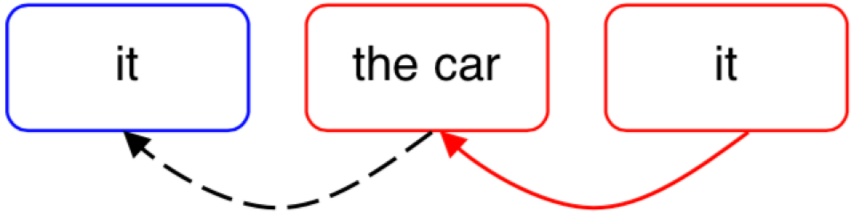
How to train a model to make local decisions such that it produces a global structure?

Some Local Decisions Matter More than Others



Severe error

*“**it** was raining, but **the car** stayed dry because **it** was under cover”*

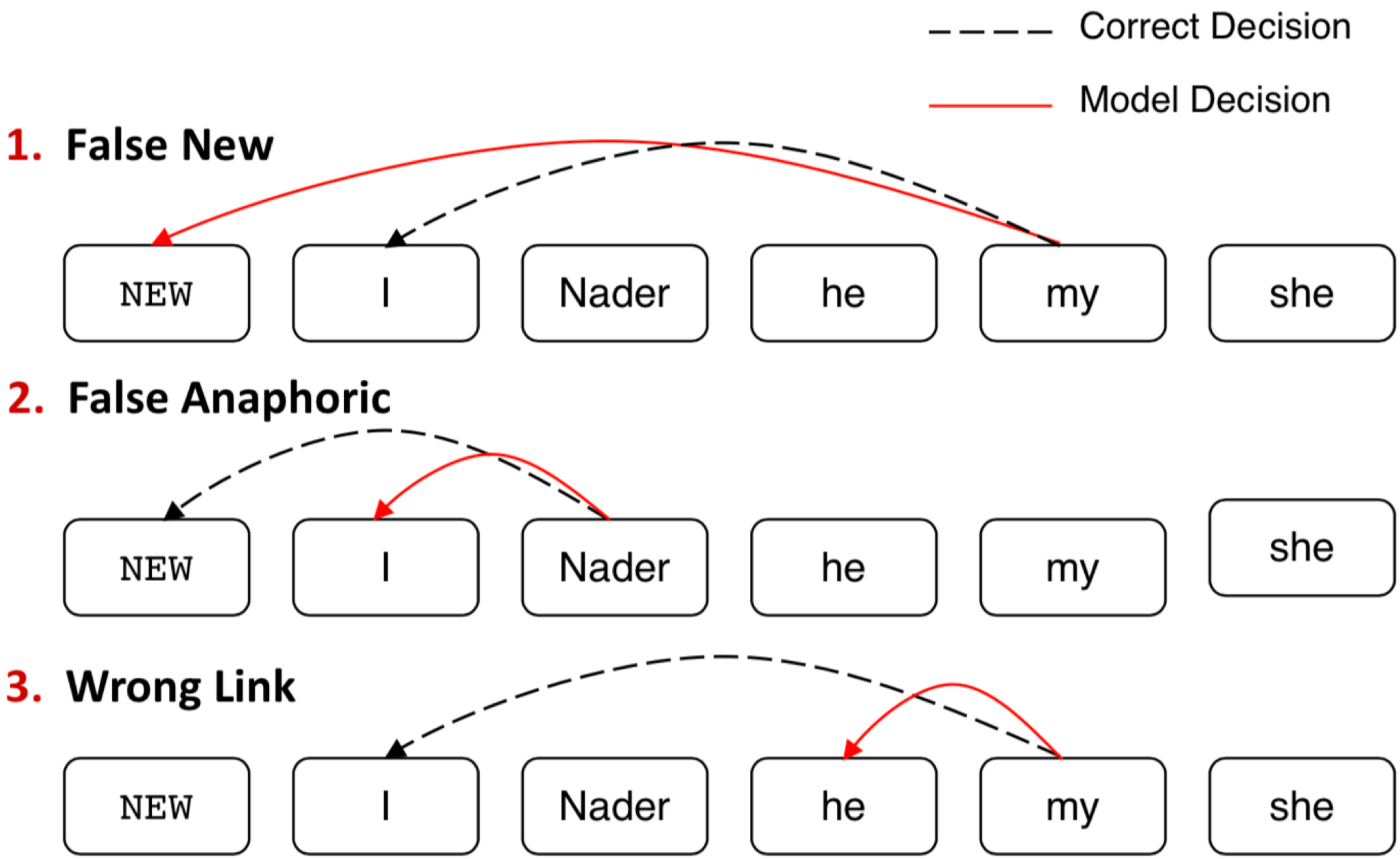


Minor error

Prior Work

Heuristically defines which error types are more important than others

Prior Work: Coreference Error Types



Learning Algorithms

Heuristic Loss Function

Prior Work: Heuristic Loss Function

Heuristically costs for mistakes

$$\Delta_h(c, m_i) = \begin{cases} 0 & \text{if } c \text{ and } m_i \text{ are coreferent} \\ \alpha_{\text{FN}} & \text{if false new error} \\ \alpha_{\text{FA}} & \text{if false anaphoric error} \\ \alpha_{\text{WL}} & \text{if wrong link error} \end{cases}$$

Prior Work: Heuristic Loss Function

Max-Margin Loss (Wiseman et al)

$$\max_{c \in \mathcal{C}(m_i)} \Delta_h(c, m_i)(1 + s(c, m_i) - s(\hat{t}_i, m_i))$$

max over candidate corefence decisions

cost for this coref decision

loss for scoring this decision too highly

Prior Work: Heuristic Loss Function

Disadvantages

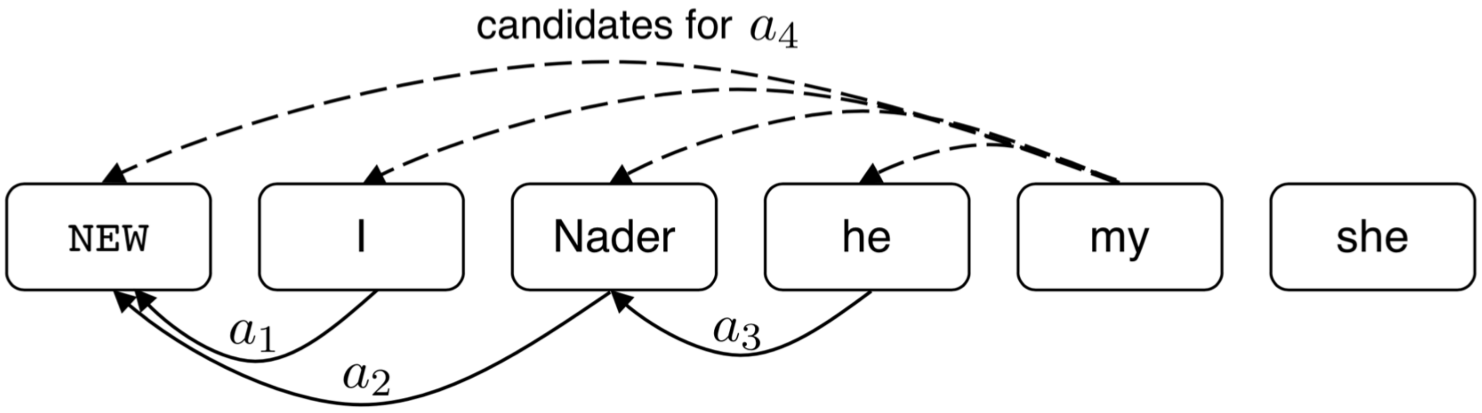
- Requires careful tuning of hyperparameters using slow grid search
- Does not generalize across datasets, languages, metrics
- Does not optimize for evaluation metric
    - At best loss is correlated with metric

Reinforcement Learning to the Rescue!

- Does not require hyperparameter training
- Small boost in accuracy

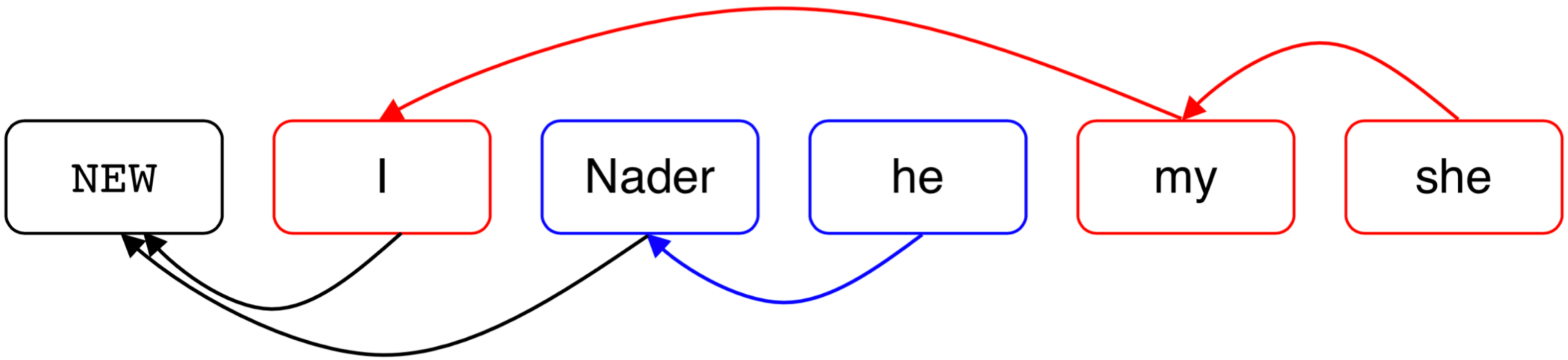
Coref Resolution with Reinforcement Learning

- Model takes a sequence of actions $a_{1:T} = a_1, a_2, \ldots, a_T$
  - action $a_i = (c, m_i)$ adds a coreference link between the i^th mention and candidate antecedent c



Coref Resolution with Reinforcement Learning

- After completing a sequence of actions, model receives a reward (B^3 metric)



$$R(a_{1:5}) = 100$$

Learning Algorithms

REINFORCE algorithm (Williams, 1992)

REINFORCE Algorithm

- Define probability distribution over actions:

  $p_\theta((c,m)) \propto e^{s(c,m)}$ for any action $a = (c,m)$

- Maximize expected reward

  $J(\theta) = \mathbb{E}_{[a_{1:T} \sim p_\theta]} R(a_{1:T})$

REINFORCE Algorithm

- Competitive with heuristic loss
- Disadvantage Vs. Max-Margin Loss
  - REINFORCE maximizes performance in expectation
    - We only need the highest scoring action(s) to be correct, not low scoring actions

Combine best of both worlds!

Improve cost-function in

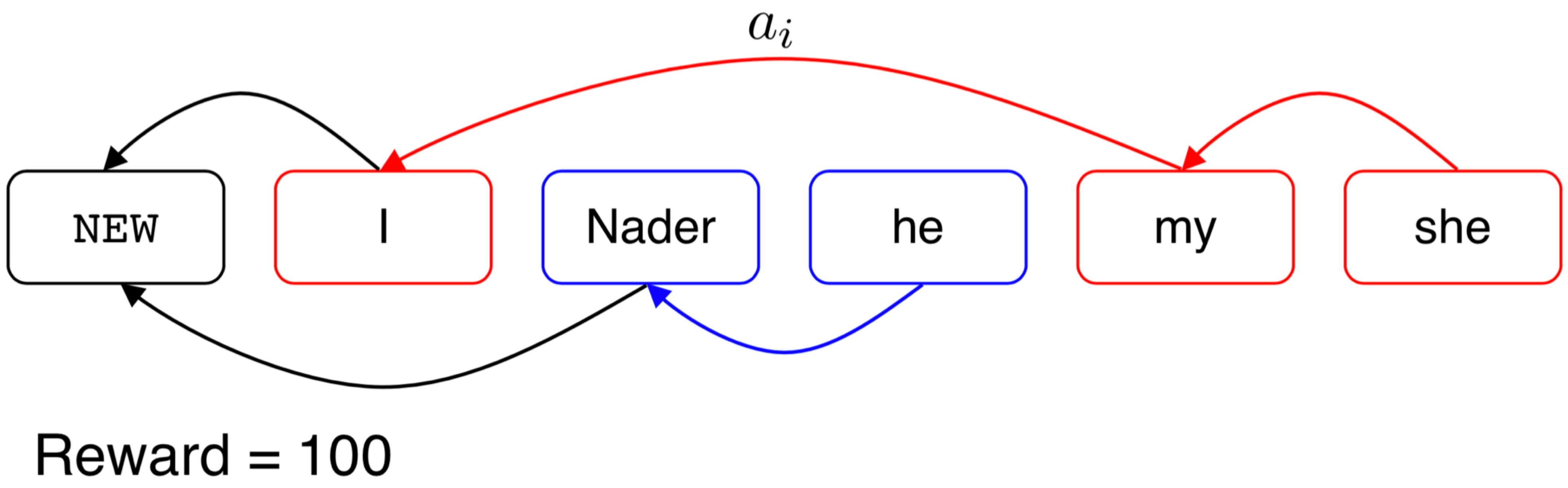
Max-Margin Loss

Learning Algorithms

Reward-Rescaling

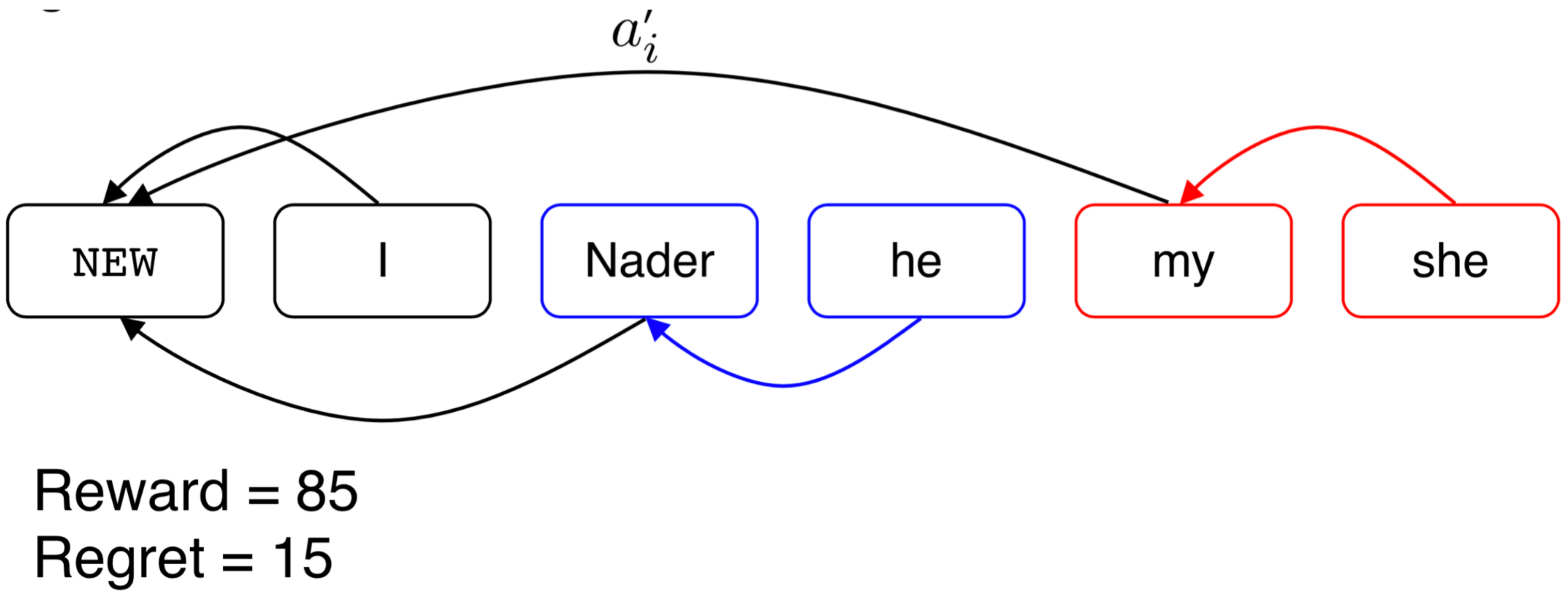
Reward-Rescaling

- Since actions are independent, we can change an action a_i to a different one a_i' and see what reward we would have gotten instead



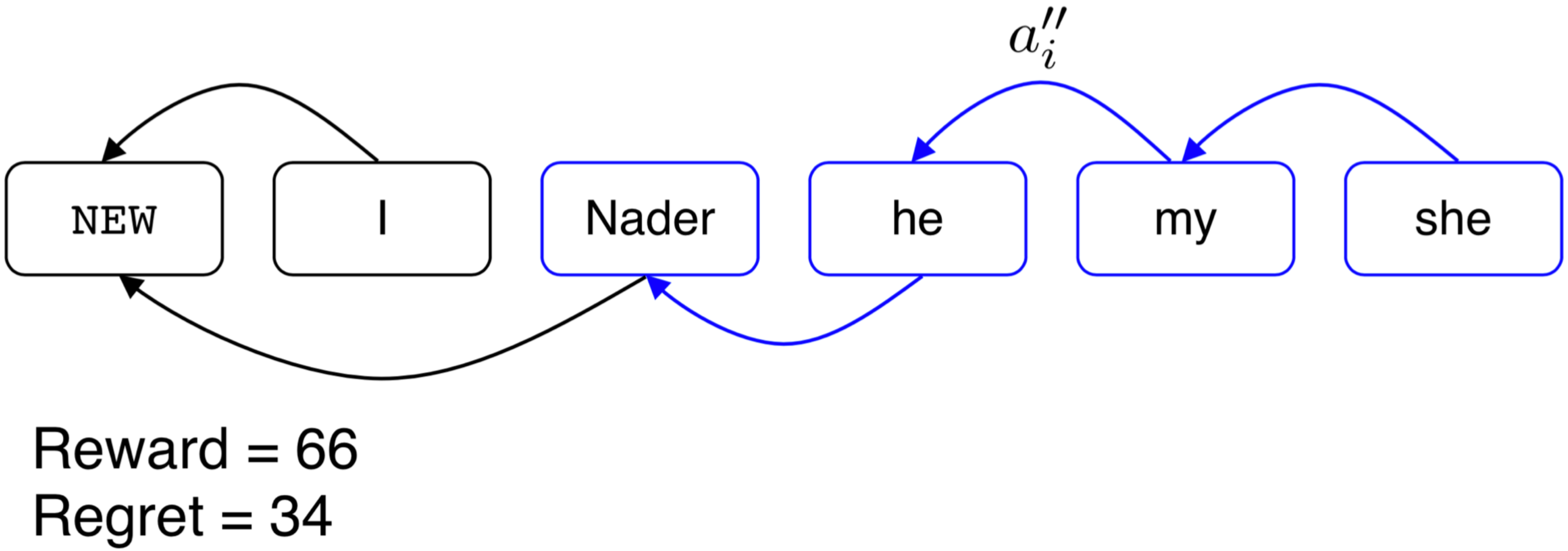
Reward-Rescaling

- Since actions are independent, we can change an action a_i to a different one a_i' and see what reward we would have gotten instead



Reward-Rescaling

- Since actions are independent, we can change an action a_i to a different one a_i' and see what reward we would have gotten instead



Reward-Rescaling

- Use this idea to do reward-based slack-rescaling

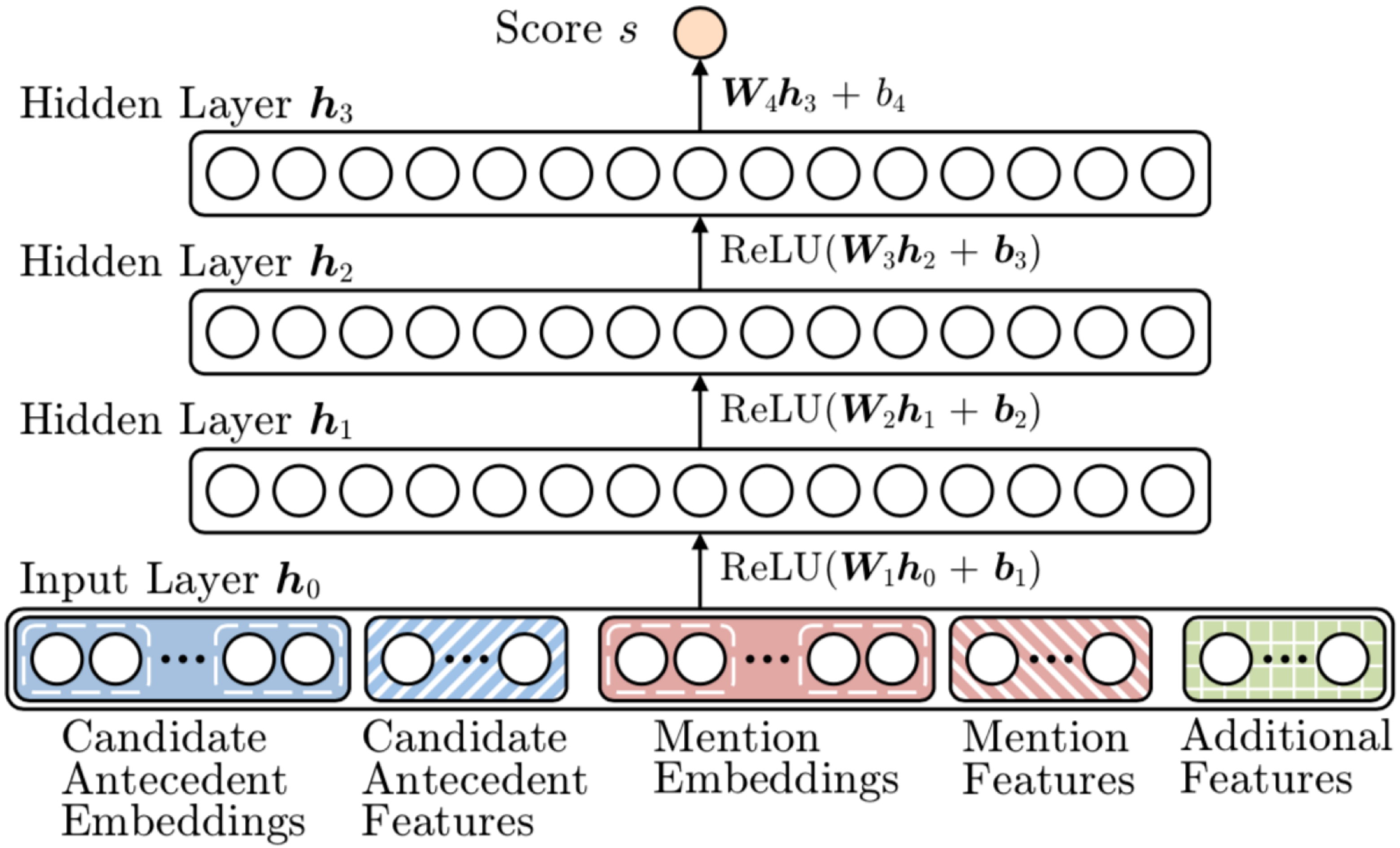
$$\Delta_r(c, m_i) = \max_{a_i' \in \mathcal{A}_i} R(a_1, ..., a_i', ..., a_T)$$ reward for best action

$$- R(a_1, ..., (c, m_i), ..., a_T)$$ reward for current action

- Cost is the regret of taking the action
  - Replaces the heuristic cost, otherwise use the same max-margin loss function

Experimental Setup

- English and Chinese CoNLL 2012 Shared Task dataset
- Mentions predicted using Stanford rule-based system (Lee et al, 2011)
- Scores are CoNLL F-1 scores
  - Average of MUC, B^3 and CEAF metrics



Neural Mention Ranking Model

Standard feed-forward neural network (Clark and Manning, 2016)

Features

- Word Embeddings
  - Previous two words, first word, last word**, head word** of each mention
  - Groups of words as average of vectors for each word in the group
- Also
  - Distance
  - String Matching
  - Document Genre
  - Speaker Information
- Separate network for anaphrocity scores

Evaluation

	MUC			B^3			$CEAF_{\phi_4}$			
	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Avg. F_1
CoNLL 2012 English Test Data										
Wiseman et al. (2016)	77.49	69.75	73.42	66.83	56.95	61.50	62.14	53.85	57.70	64.21
Clark & Manning (2016)	79.91	69.30	74.23	71.01	56.53	62.95	63.84	54.33	58.70	65.29
Heuristic Loss	79.63	70.25	74.65	69.21	57.87	63.03	63.62	53.97	58.40	65.36
REINFORCE	80.08	69.61	74.48	70.70	56.96	63.09	63.59	54.46	58.67	65.41
Reward Rescaling	79.19	70.44	74.56	69.93	57.99	63.40	63.46	55.52	59.23	65.73
CoNLL 2012 Chinese Test Data										
Björkelund & Kuhn (2014)	69.39	62.57	65.80	61.64	53.87	57.49	59.33	54.65	56.89	60.06
Clark & Manning (2016)	74.45	64.73	69.25	68.71	55.54	61.43	63.14	57.48	60.18	63.62
Heuristic Loss	72.20	66.51	69.24	64.71	58.16	61.26	61.98	58.41	60.14	63.54
REINFORCE	74.05	65.38	69.44	67.52	56.43	61.48	62.38	57.77	59.98	63.64
Reward Rescaling	73.64	65.62	69.40	67.48	56.94	61.76	62.46	58.60	60.47	63.88

Table 1: Comparison of the methods together with other state-of-the-art approaches on the test sets.

Error Breakdown: Avoiding Costly Mistakes

- Reward-Rescaling makes more errors in total!
  - However, the errors are less severe

Model	FN	FA	WL
Heuristic Loss	1719	1956	1258
Reward Rescaling	1725	1994	1247

Table 2: Number of "false new," "false anaphoric," and "wrong link" errors produced by the models on the English CoNLL 2012 test set.

Comparison with Heuristic Loss

- High variance in costs for a given error type
  - Distribution of "False New" cost is spread out, so using fixed penalty for an error-type is insufficient

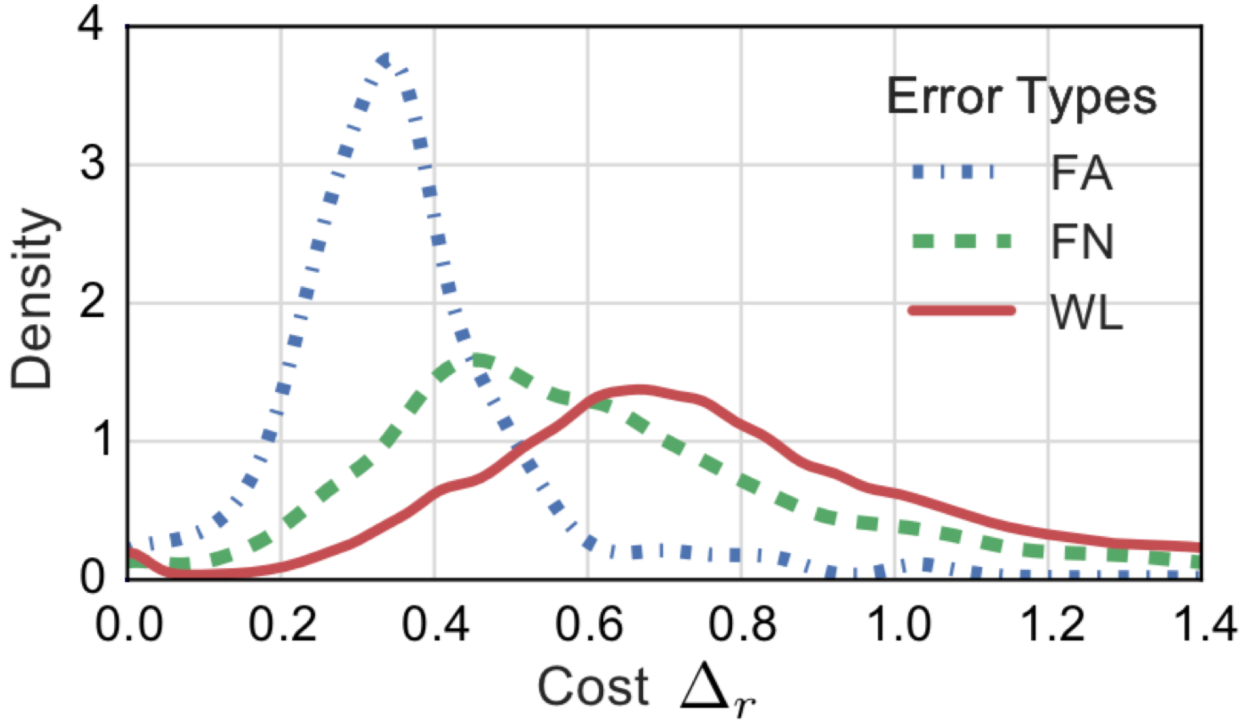


Figure 1: Density plot of the costs Δ_r associated with different error types on the English CoNLL 2012 test set.

Example Improvement: Proper Nouns

- Fewer "false new" errors with proper nouns

Class of Mentions	Average Cost $\overline{\Delta}_r$			# Heuristic Loss Errors			# Reward Rescaling Errors		
	FN	FA	WL	FN	FA	WL	FN	FA	WL
Proper nouns	0.90	0.38	1.02	403	597	221	334	660	233
Pronouns in phone conversations	0.86	0.39	1.21	82	85	81	90	78	67

Table 3: Examples of classes of mention on which the reward-rescaling loss significantly improves upon the heuristic loss due to its reward-based cost function. Reported numbers are from the English CoNLL 2012 test set.

Conclusion

Heuristic Loss < REINFORCE < Reward-Rescaling

- Why?
  - Benefit of Max-Margin Loss
  - Directly optimizes coref metrics rather than heuristic cost function

- Advantages:
  - Does not require hyperparameter training
  - Small boost in accuracy with fewer costly mistakes

Caveats

- Reward metric needs to be fast since it will be computed many times!
- May overfit for evaluation metric

Thank You

Any Questions?