Deep Reinforcement Learning for Mention-Ranking Coreference Models

Kevin Clark
Computer Science Department
Stanford University
kevclark@cs.stanford.edu

Christopher D. Manning
Computer Science Department
Stanford University
manning@cs.stanford.edu

Abstract

Coreference resolution systems are typically trained with heuristic loss functions that require careful tuning. In this paper we instead apply reinforcement learning to directly optimize a neural mention-ranking model for coreference evaluation metrics. We experiment with two approaches: the REINFORCE policy gradient algorithm and a reward-rescaled max-margin objective. We find the latter to be more effective, resulting in significant improvements over the current state-of-the-art on the English and Chinese portions of the CoNLL 2012 Shared Task.

1 Introduction

Coreference resolution systems typically operate by making sequences of local decisions (e.g., adding a coreference link between two mentions). However, most measures of coreference resolution performance do not decompose over local decisions, which means the utility of a particular decision is not known until all other decisions have been made.

Due to this difficulty, coreference systems are usually trained with loss functions that heuristically define the goodness of a particular coreference decision. These losses contain hyperparameters that are carefully selected to ensure the model performs well according to coreference evaluation metrics. This complicates training, especially across different languages and datasets where systems may work best with different settings of the hyperparameters.

To address this, we explore using two variants of reinforcement learning to directly optimize a coreference system for coreference evaluation metrics. In particular, we modify the max-margin coreference objective proposed by Wiseman et al. (2015) by incorporating the reward associated with each coreference decision into the loss's slack rescaling. We also test the REINFORCE policy gradient algorithm (Williams, 1992).

Our model is a neural mention-ranking model. Mention-ranking models score pairs of mentions for their likelihood of coreference rather than comparing partial coreference clusters. Hence they operate in a simple setting where coreference decisions are made independently. Although they are less expressive than entity-centric approaches to coreference (e.g., Haghighi and Klein, 2010), mention-ranking models are fast, scalable, and simple to train, causing them to be the dominant approach to coreference in recent years (Durrett and Klein, 2013; Wiseman et al., 2015). Having independent actions is particularly useful when applying reinforcement learning because it means a particular action's effect on the final reward can be computed efficiently.

We evaluate the models on the English and Chinese portions of the CoNLL 2012 Shared Task. The REINFORCE algorithm is competitive with a heuristic loss function while the reward-rescaled objective significantly outperforms both[1]. We attribute this to reward rescaling being well suited for a ranking task due to its max-margin loss as well as benefiting from directly optimizing for coreference metrics. Error analysis shows that using the reward-rescaling loss results in a similar number of mistakes as the heuristic loss, but the mistakes tend to be less severe.

[1]Code and trained models are available at https://github.com/clarkkev/deep-coref.

2 Neural Mention-Ranking Model

We use the neural mention-ranking model described in Clark and Manning (2016), which we briefly go over in this section. Given a mention m and candidate antecedent c, the mention-ranking model produces a score for the pair $s(c, m)$ indicating their compatibility for coreference with a feedforward neural network. The candidate antecedent may be any mention that occurs before m in the document or NA, indicating that m has no antecedent.

Input Layer. For each mention, the model extracts various words (e.g., the mention's head word) and groups of words (e.g., all words in the mention's sentence) that are fed into the neural network. Each word is represented by a vector $\boldsymbol{w}_i \in \mathbb{R}^{d_w}$. Each group of words is represented by the average of the vectors of each word in the group. In addition to the embeddings, a small number of additional features are used, including distance, string matching, and speaker identification features. See Clark and Manning (2016) for the full set of features and an ablation study.

These features are concatenated to produce an I-dimensional vector $\boldsymbol{h}_0$, the input to the neural network. If $c = \text{NA}$, features defined over pairs of mentions are not included. For this case, we train a separate network with an identical architecture to the pair network except for the input layer to produce anaphoricity scores.

Hidden Layers. The input gets passed through three hidden layers of rectified linear (ReLU) units (Nair and Hinton, 2010). Each unit in a hidden layer is fully connected to the previous layer:

$$\boldsymbol{h}_i(c, m) = \max(0, \boldsymbol{W}_i \boldsymbol{h}_{i-1}(c, m) + \boldsymbol{b}_i)$$

where $\boldsymbol{W}_1$ is a $M_1 \times I$ weight matrix, $\boldsymbol{W}_2$ is a $M_2 \times M_1$ matrix, and $\boldsymbol{W}_3$ is a $M_3 \times M_2$ matrix.

Scoring Layer. The final layer is a fully connected layer of size 1:

$$s(c, m) = \boldsymbol{W}_4 \boldsymbol{h}_3(c, m) + b_4$$

where $\boldsymbol{W}_4$ is a $1 \times M_3$ weight matrix. At test time, the mention-ranking model links each mention with its highest scoring candidate antecedent.

3 Learning Algorithms

Mention-ranking models are typically trained with heuristic loss functions that are tuned via hyperparameters. These hyperparameters are usually given as costs for different error types, which are used to bias the coreference system towards making more or fewer coreference links.

In this section we first describe a heuristic loss function incorporating this idea from Wiseman et al. (2015). We then propose new training procedures based on reinforcement learning that instead directly optimize for coreference evaluation metrics.

3.1 Heuristic Max-Margin Objective

The heuristic loss from Wiseman et al. is governed by the following error types, which were first proposed by Durrett et al. (2013).

Suppose the training set consists of N mentions $m_1, m_2, ..., m_N$. Let $\mathcal{C}(m_i)$ denote the set of candidate antecedents of a mention m_i (i.e., mentions preceding m_i and NA) and $\mathcal{T}(m_i)$ denote the set of true antecedents of m_i (i.e., mentions preceding m_i that are coreferent with it or $\{\text{NA}\}$ if m_i has no antecedent). Then we define the following costs for linking m_i to a candidate antecedent $c \in \mathcal{C}(m_i)$:

$$\Delta_h(c, m_i) = \begin{cases} \alpha_{\text{FN}} & \text{if } c = \text{NA} \ \wedge \mathcal{T}(m_i) \neq \{\text{NA}\} \\ \alpha_{\text{FA}} & \text{if } c \neq \text{NA} \ \wedge \mathcal{T}(m_i) = \{\text{NA}\} \\ \alpha_{\text{WL}} & \text{if } c \neq \text{NA} \ \wedge c \notin \mathcal{T}(m_i) \\ 0 & \text{if } c \in \mathcal{T}(m_i) \end{cases}$$

for "false new," "false anaphor," "wrong link", and correct coreference decisions.

The heuristic loss is a slack-rescaled max-margin objective parameterized by these error costs. Let $\hat{t}_i$ be the highest scoring true antecedent of m_i:

$$\hat{t}_i = \underset{c \in \mathcal{C}(m_i) \wedge \Delta_h(c, m_i) = 0}{\text{argmax}} s(c, m_i)$$

Then the heuristic loss is given as

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \max_{c \in \mathcal{C}(m_i)} \Delta_h(c, m_i)(1 + s(c, m_i) - s(\hat{t}_i, m_i))$$

Finding Effective Error Penalties. We fix $\alpha_{\text{WL}} = 1.0$ and search for α_{FA} and α_{FN} out of $\{0.1, 0.2, ..., 1.5\}$ with a variant of grid search. Each new trial uses the unexplored set of hyperparame-

ters that has the closest Manhattan distance to the best setting found so far on the dev set. The search is halted when all immediate neighbors (within 0.1 distance) of the best setting have been explored. We found $(\alpha_{\text{FN}}, \alpha_{\text{FA}}, \alpha_{\text{WL}}) = (0.8, 0.4, 1.0)$ to be best for English and $(\alpha_{\text{FN}}, \alpha_{\text{FA}}, \alpha_{\text{WL}}) = (0.8, 0.5, 1.0)$ to be best for Chinese on the CoNLL 2012 data.

3.2 Reinforcement Learning

Finding the best hyperparameter settings for the heuristic loss requires training many variants of the model, and at best results in an objective that is correlated with coreference evaluation metrics. To address this, we pose mention ranking in the reinforcement learning framework (Sutton and Barto, 1998) and propose methods that directly optimize the model for coreference metrics.

We can view the mention-ranking model as an *agent* taking a series of *actions* $a_{1:T} = a_1, a_2, ..., a_T$, where T is the number of mentions in the current document. Each action a_i links the ith mention in the document m_i to a candidate antecedent. Formally, we denote the set of actions available for the ith mention as $\mathcal{A}_i = \{(c, m_i) : c \in \mathcal{C}(m_i)\}$, where an action (c, m) adds a coreference link between mentions c and m. The mention-ranking model assigns each action the score $s(c, m)$ and takes the highest-scoring action at each step.

Once the agent has executed a sequence of actions, it observes a *reward* $R(a_{1:T})$, which can be any function. We use the B^3 coreference metric for this reward (Bagga and Baldwin, 1998). Although our system evaluation also includes the MUC (Vilain et al., 1995) and CEAF_{ϕ_4} (Luo, 2005) metrics, we do not incorporate them into the loss because MUC has the flaw of treating all errors equally and CEAF_{ϕ_4} is slow to compute.

Reward Rescaling. Crucially, the actions taken by a mention-ranking model are independent. This means it is possible to change any action a_i to a different one $a'_i \in \mathcal{A}_i$ and see what reward the model would have gotten by taking that action instead: $R(a_1, ..., a_{i-1}, a'_i, a_{i+1}, ..., a_T)$. We use this idea to improve the slack-rescaling parameter Δ in the max-margin loss $\mathcal{L}(\theta)$. Instead of setting its value based on the error type, we compute exactly how much each action hurts the final reward:

$$\Delta_r(c, m_i) = -R(a_1, ..., (c, m_i), ..., a_T) + \max_{a'_i \in \mathcal{A}_i} R(a_1, ..., a'_i, ..., a_T)$$

where $a_{1:T}$ is the highest scoring sequence of actions according to the model's current parameters. Otherwise the model is trained in the same way as with the heuristic loss.

The REINFORCE Algorithm. We also explore using the REINFORCE policy gradient algorithm (Williams, 1992). We can define a probability distribution over actions using the mention-ranking model's scoring function as follows:

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

for any action $a = (c, m)$. The REINFORCE algorithm seeks to maximize the expected reward

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

It does this through gradient ascent. Computing the full gradient is prohibitive because of the expectation over all possible action sequences, which is exponential in the length of the sequence. Instead, it gets an unbiased estimate of the gradient by sampling a sequence of actions $a_{1:T}$ according to p_θ and computing the gradient only over the sample.

We take advantage of the independence of actions by using the following gradient estimate, which has lower variance than the standard REINFORCE gradient estimate:

$$\nabla_\theta J(\theta) \approx \sum_{i=1}^{T} \sum_{a'_i \in \mathcal{A}_i} [\nabla_\theta p_\theta(a'_i)](R(a_1, ..., a'_i, ..., a_T) - b_i)$$

where b_i is a baseline used to reduce the variance, which we set to $\mathbb{E}_{a'_i \in \mathcal{A}_i \sim p_\theta} R(a_1, ..., a'_i, ..., a_T)$.

4 Experiments and Results

We run experiments on the English and Chinese portions of the CoNLL 2012 Shared Task data (Pradhan et al., 2012) and evaluate with the MUC, B^3, and CEAF_{ϕ_4} metrics. Our experiments were run using predicted mentions from Stanford's rule-based coreference system (Raghunathan et al., 2010).

We follow the training methodology from Clark and Manning (2016): hidden layers of sizes M_1 = 1000, M_2 = M_3 = 500, the RMSprop optimizer

	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)	73.85	65.42	69.38	67.53	56.41	61.47	62.84	57.62	60.12	63.66
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.

(Hinton and Tieleman, 2012), dropout (Hinton et al., 2012) with a rate of 0.5, and pretraining with the all pairs classification and top pairs classification tasks. However, we improve on the previous system by using using better mention detection, more effective hyperparameters, and more epochs of training.

4.1 Results

We compare the heuristic loss, REINFORCE, and reward rescaling approaches on both datasets. We find that REINFORCE does slightly better than the heuristic loss, but reward rescaling performs significantly better than both on both languages.

We attribute the modest improvement of REINFORCE to it being poorly suited for a ranking task. During training it optimizes the model's performance in expectation, but at test-time it takes the most probable sequence of actions. This mismatch occurs even at the level of an individual decision: the model only links the current mention to a single antecedent, but is trained to assign high probability to all correct antecedents. We believe the benefit of REINFORCE being guided by coreference evaluation metrics is offset by this disadvantage, which does not occur in the max-margin approaches. The reward-rescaled max-margin loss combines the best of both worlds, resulting in superior performance.

4.2 The Benefits of Reinforcement Learning

In this section we examine the reward-based cost function Δ_r and perform error analysis to determine how reward rescaling improves the mention-ranking model's accuracy.

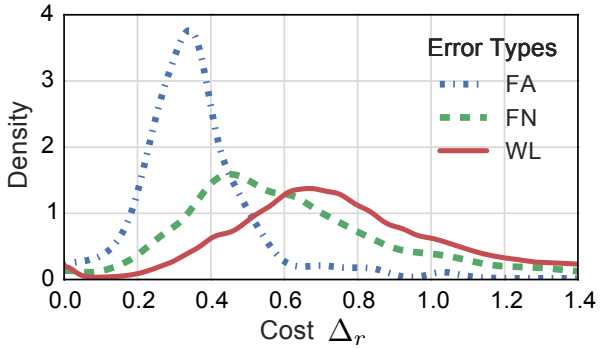

Figure 1: Density plot of the costs Δ_r associated with different error types on the English CoNLL 2012 test set.

Comparison with Heuristic Costs. We compare the reward-based cost function Δ_r with the error types used in the heuristic loss. For English, the average value of Δ_r is 0.79 for FN errors and 0.38 for FA errors when the costs are scaled so the average value of a WL error is 1.0. These are very close to the hyperparameter values $(\alpha_{\text{FN}}, \alpha_{\text{FA}}, \alpha_{\text{WL}}) = (0.8, 0.4, 1.0)$ found by grid search. However, there is a high variance in costs for each error type, suggesting that using a fixed penalty for each type as in the heuristic loss is insufficient (see Figure 1).

Avoiding Costly Mistakes. Embedding the costs of actions into the loss function causes the reward-rescaling model to prioritize getting the more important coreference decisions (i.e., the ones with the biggest impact on the final score) correct. As a

Mention Type	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 improves upon the heuristic loss due to its reward-based cost function. Reported numbers are from the English CoNLL 2012 test set.

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.

result, it makes fewer costly mistakes at test time. Costly mistakes often involve large clusters of mentions: incorrectly combining two coreference clusters of size ten is much worse than incorrectly combining two clusters of size one. However, the cost of an action also depends on other factors such as the number of errors already present in the clusters and the utilities of the other available actions.

Table 2 shows the breakdown of errors made by the heuristic and reward-rescaling models on the test set. The reward-rescaling model makes slightly more errors, meaning its improvement in performance must come from its errors being less severe.

Example Improvements. Table 3 shows two classes of mentions where the reward-rescaling loss particularly improves over the heuristic loss.

Proper nouns have a higher average cost for "false new" errors (0.90) than other mentions types (0.77). This is perhaps because proper nouns are important for connecting clusters of mentions far apart in a document, so incorrectly linking a proper noun to NA could result in a large decrease in recall. Because it more heavily weights these high-cost errors during training, the reward-rescaling model makes fewer "false new" errors for proper nouns than the heuristic loss. Although there is an increase in other kinds of errors as a result, most of these are low-cost "false anaphoric" errors.

The pronouns in the "telephone conversation" genre often group into extremely large coreference clusters, which means a "wrong link" error can have a large negative effect on the score. This is reflected in its high average cost of 1.21. After prioritizing these examples during training, the reward-rescaling model creates significantly fewer wrong links than the heuristic loss, which is trained using a fixed cost of 1.0 for all wrong links.

5 Related Work

Mention-ranking models have been widely used for coreference resolution (Denis and Baldridge, 2007; Rahman and Ng, 2009; Durrett and Klein, 2013). These models are typically trained with heuristic loss functions that assign costs to different error types, as in the heuristic loss we describe in Section 3.1 (Fernandes et al., 2012; Durrett et al., 2013; Björkelund and Kuhn, 2014; Wiseman et al., 2015; Martschat and Strube, 2015; Wiseman et al., 2016).

To the best of our knowledge reinforcement learning has not been applied to coreference resolution before. However, imitation learning algorithms such as SEARN (Daumé III et al., 2009) have been used to train coreference resolvers (Daumé III, 2006; Ma et al., 2014; Clark and Manning, 2015). These algorithms also directly optimize for coreference evaluation metrics, but they require an expert policy to learn from instead of relying on rewards alone.

6 Conclusion

We propose using reinforcement learning to directly optimize mention-ranking models for coreference evaluation metrics, obviating the need for hyperparameters that must be carefully selected for each particular language, dataset, and evaluation metric. Our reward-rescaling approach also increases the model's accuracy, resulting in significant gains over the current state-of-the-art.

Acknowledgments

We thank Kelvin Guu, William Hamilton, Will Monroe, and the anonymous reviewers for their thoughtful comments and suggestions. This work was supported by NSF Award IIS-1514268.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566.

Anders Björkelund and Jonas Kuhn. 2014. Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *Association of Computational Linguistics (ACL)*.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. In *Association for Computational Linguistics (ACL)*.

Kevin Clark and Christopher D. Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. In *Association for Computational Linguistics (ACL)*.

Hal Daumé III, John Langford, and Daniel Marcu. 2009. Search-based structured prediction. *Machine Learning*, 75(3):297–325.

Hal Daumé III. 2006. *Practical structured learning techniques for natural language processing*. Ph.D. thesis, University of Southern California, Los Angeles, CA.

Pascal Denis and Jason Baldridge. 2007. A ranking approach to pronoun resolution. In *International Joint Conferences on Artificial Intelligence (IJCAI)*.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Greg Durrett, David Leo Wright Hall, and Dan Klein. 2013. Decentralized entity-level modeling for coreference resolution. In *Association for Computational Linguistics (ACL)*.

Eraldo Rezende Fernandes, Cícero Nogueira Dos Santos, and Ruy Luiz Milidiú. 2012. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning - Shared Task*, pages 41–48.

Aria Haghighi and Dan Klein. 2010. Coreference resolution in a modular, entity-centered model. In *Human Language Technology and North American Association for Computational Linguistics (HLT-NAACL)*.

Geoffrey Hinton and Tijmen Tieleman. 2012. Lecture 6.5-RmsProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning, 4*.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Chao Ma, Janardhan Rao Doppa, J Walker Orr, Prashanth Mannem, Xiaoli Fern, Tom Dietterich, and Prasad Tadepalli. 2014. Prune-and-score: Learning for greedy coreference resolution. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Sebastian Martschat and Michael Strube. 2015. Latent structures for coreference resolution. *Transactions of the Association for Computational Linguistics (TACL)*, 3:405–418.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning (ICML)*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning - Shared Task*, pages 1–40.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Altaf Rahman and Vincent Ng. 2009. Supervised models for coreference resolution. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Richard S Sutton and Andrew G Barto. 1998. *Reinforcement learning: An introduction*. MIT Press.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on message understanding*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Sam Wiseman, Alexander M. Rush, Stuart M. Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Association of Computational Linguistics (ACL)*.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Human Language Technology and North American Association for Computational Linguistics (HLT-NAACL)*.