End-to-end Deep Reinforcement Learning Based Coreference Resolution

Hongliang Fei, Xu Li, Dingcheng Li, Ping Li

Cognitive Computing Lab, Baidu Research
{hongliangfei,lixu13,lidingcheng,liping11}@baidu.com

Abstract

Recent neural network models have significantly advanced the task of coreference resolution. However, current neural coreference models are typically trained with heuristic loss functions that are computed over a sequence of local decisions. In this paper, we introduce an end-to-end reinforcement learning based coreference resolution model to directly optimize coreference evaluation metrics. Specifically, we modify the state-of-the-art higher-order mention ranking approach in Lee et al. (2018) to a reinforced policy gradient model by incorporating the reward associated with a sequence of coreference linking actions. Furthermore, we introduce maximum entropy regularization for adequate exploration to prevent the model from prematurely converging to a bad local optimum. Our proposed model achieves new state-of-the-art performance on the English OntoNotes v5.0 benchmark.

1 Introduction

Coreference resolution is one of the most fundamental tasks in natural language processing (NLP), which has a significant impact on many downstream applications including information extraction (Dai et al., 2019), question answering (Weston et al., 2015), and entity linking (Hajishirzi et al., 2013). Given an input text, coreference resolution aims to identify and group all the mentions that refer to the same entity.

In recent years, deep neural network models for coreference resolution have been prevalent (Wiseman et al., 2016; Clark and Manning, 2016b). These models, however, either assumed mentions were given and only developed a coreference linking model (Clark and Manning, 2016b) or built a pipeline system to detect mention first then resolved coreferences (Haghighi and Klein, 2010). In either case, they depend on hand-crafted features and syntactic parsers that may not generalize well or may even propagate errors.

To avoid the cascading errors of pipeline systems, recent NLP researchers have developed end-to-end approaches (Lee et al., 2017; Luan et al., 2018; Lee et al., 2018; Zhang et al., 2018), which directly consider all text spans, jointly identify entity mentions and cluster them. The core of those end-to-end models are vector embeddings to represent text spans in the document and scoring functions to compute the mention scores for text spans and antecedent scores for pairs of spans. Depending on how the span embeddings are computed, the end-to-end coreference models could be further divided into first order methods (Lee et al., 2017; Luan et al., 2018; Zhang et al., 2018) or higher order methods (Lee et al., 2018).

Although recent end-to-end neural coreference models have advanced the state-of-the-art performance for coreference resolution, they are still trained with heuristic loss functions and make a sequence of local decisions for each pair of mentions. However as studied in Clark and Manning (2016a); Yin et al. (2018), most coreference resolution evaluation measures are not accessible over local decisions, but can only be known until all other decisions have been made. Therefore, the next key research question is how to integrate and directly optimize coreference evaluation metrics in an end-to-end manner.

In this paper, we propose a goal-directed end-to-end deep reinforcement learning framework to resolve coreference as shown in Figure 1. Specifically, we leverage the neural architecture in Lee et al. (2018) as our policy network, which includes learning span representation, scoring potential entity mentions, and generating a probability distribution over all possible coreference linking actions from the current mention to its antecedents. Once a sequence of linking actions are made, our

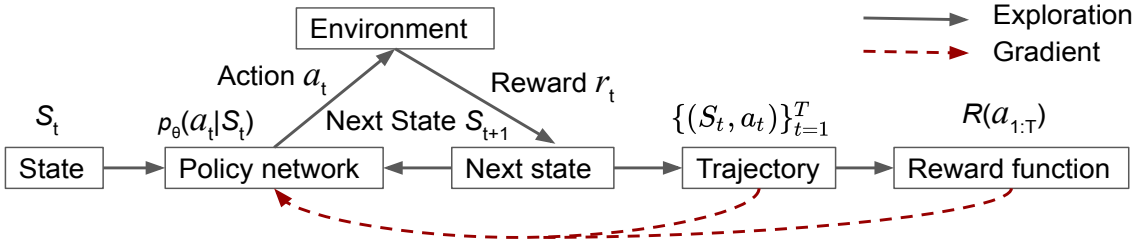

Figure 1: The basic framework of our policy gradient model for one trajectory. The policy network is an end-to-end neural module that can generate probability distributions over actions of coreference linking. The reward function computes a reward given a trajectory of actions based on coreference evaluation metrics. Solid line indicates the model exploration and (red) dashed line indicates the gradient update.

reward function is used to measure how good the generated coreference clusters are, which is directly related to coreference evaluation metrics. Besides, we introduce an entropy regularization term to encourage exploration and prevent the policy from prematurely converging to a bad local optimum. Finally, we update the regularized policy network parameters based on the rewards associated with sequences of sampled actions, which are computed on the whole input document.

We evaluate our end-to-end reinforced coreference resolution model on the English OntoNotes v5.0 benchmark. Our model achieves the new state-of-the-art F1-score of 73.8%, which outperforms previous best-published result (73.0%) of Lee et al. (2018) with statistical significance.

2 Related Work

Closely related to our work are the end-to-end coreference models developed by Lee et al. (2017) and Lee et al. (2018). Different from previous pipeline approaches, Lee et al. used neural networks to learn mention representations and calculate mention and antecedent scores without using syntactic parsers. However, their models optimize a heuristic loss based on local decisions rather than the actual coreference evaluation metrics, while our reinforcement model directly optimizes the evaluation metrics based on the rewards calculated from sequences of actions.

Our work is also inspired by Clark and Manning (2016a) and Yin et al. (2018), which resolve coreferences with reinforcement learning techniques. They view the mention-ranking model as an agent taking a series of actions, where each action links each mention to a candidate antecedent. They also use pretraining for initialization. Nevertheless, their models assume mentions are given while our work is end-to-end. Furthermore, we add entropy regularization to encourage more exploration (Mnih et al.; Eysenbach et al., 2019) and prevent our model from prematurely converging to a sub-optimal (or bad) local optimum.

3 Methodology

3.1 Task definition

Given a document, the task of end-to-end coreference resolution aims to identify a set of mention clusters, each of which refers to the same entity. Following Lee et al. (2017), we formulate the task as a sequence of linking decisions for each span i to the set of its possible antecedents, denoted as $\mathcal{Y}(i) = \{\epsilon, 1, \cdots, i-1\}$, a dummy antecedent ϵ and all preceding spans. In particular, the use of dummy antecedent ϵ for a span is to handle two possible scenarios: (i) the span is not an entity mention or (ii) the span is an entity mention but it is not coreferent with any previous spans. The final coreference clusters can be recovered with a backtracking step on the antecedent predictions.

3.2 Our Model

Figure 2 illustrates a demonstration of our iterative coreference resolution model on a document. Given a document, our model first identifies top scored mentions, and then conducts a sequence of actions $a_{1:T} = \{a_1, a_2, \cdots, a_T\}$ over them, where T is the number of mentions and each action a_t assigns mention t to a candidate antecedent y_t in $\mathcal{Y}_t = \{\epsilon, 1, \cdots, t-1\}$. The state at time t is defined as $S_t = \{\mathbf{g}_1, \cdots, \mathbf{g}_{t-1}, \mathbf{g}_t\}$, where $\mathbf{g}_i$ is the mention i's representation.

Once our model has finished all the actions, it observes a reward $R(a_{1:T})$. The calculated gradients are then propagated to update model parameters. We use the average of the three metrics: MUC (Grishman and Sundheim, 1995), B^3 (Recasens and Hovy, 2011) and CEAFϕ_4 (Cai and

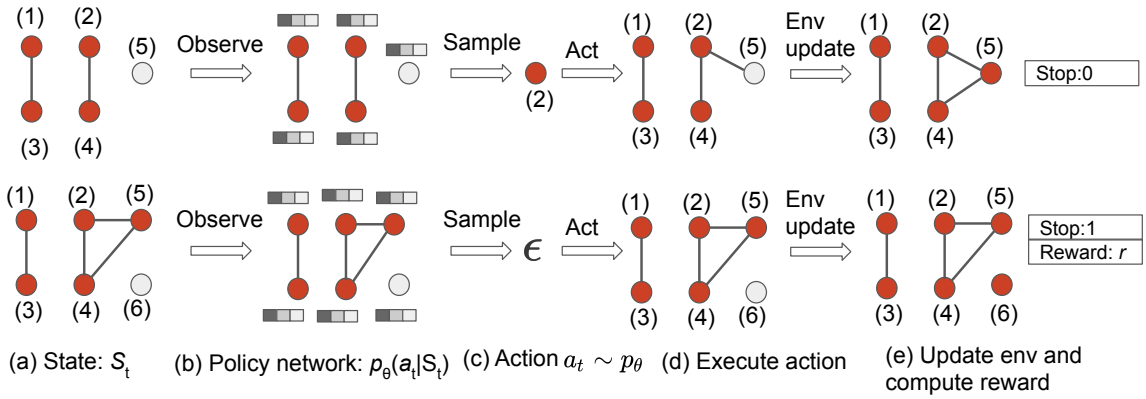

Figure 2: A demonstration of our reinforced coreference resolution method on a document with 6 mentions. The upper and lower rows correspond to step 5 and 6 respectively, in which the policy network selects mention (2) as the antecedent of mention (5) and leaves mention (6) as a singleton mention. The red (gray) nodes represent processed (current) mentions and edges between them indicate current predicted coreferential relations. The gray rectangles around circles are span embeddings and the reward is calculated at the trajectory end.

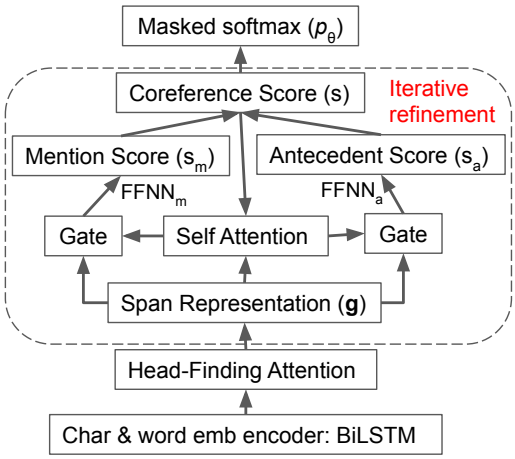

Figure 3: Architecture of the policy network. The components in dashed square iteratively refine span representations. The last layer is a masked softmax layer that computes probability distribution only over the candidate antecedents for each mention. We omit the span generation and pruning component for simplicity.

Strube, 2010) as the reward. Following Clark and Manning (2016a), we assume actions are independent and the next state S_{t+1} is generated based on the natural order of the starting position and then the end position of mentions regardless of action a_t.

Policy Network: We adopt the state-of-the-art end-to-end neural coreferene scoring architecture from Lee et al. (2018) and add a masked softmax layer to compute the probability distribution over actions, as illustrated in Figure 3. The success of their approach lies in two aspects: (i) a coarse-to-fine pruning to reduce the search space, and (ii) an iterative procedure to refine the span representation with an self-attention mechanism that averages over the previous round's representations weighted by the normalized coreference scores.

Given the state S_t and current network parameters θ, the probability of action a_t choosing y_t is:

$$p_\theta(a_t = y_t|S_t) = \frac{\exp\left(s(t, y_t)\right)}{\sum_{y' \in \mathcal{Y}_t} \exp\left(s(t, y')\right)} \quad (1)$$

where $s(i, j)$ is the pairwise coreference score between span i and span j defined as following:

$$s(i, j) = s_m(i) + s_m(j) + s_c(i, j) + s_a(i, j) \quad (2)$$

For the dummy antecedent, the score $s(i, \epsilon)$ is fixed to 0. Here $s_m(.)$ is the mention score function, $s_c(.,.)$ is a bilinear score function used to prune antecedents, and $s_a(.,.)$ is the antecedent score function. Let $\mathbf{g}_i$ denote the refined representation for span i after gating, the three functions are $s_m(i) = \theta_m^T \text{FFNN}_m(\mathbf{g}_i)$, $s_c(i, j) = \mathbf{g}_i^T \Theta_c \mathbf{g}_j$, and $s_a(i, j)$ is:

$$s_a(i, j) = \theta_a^T \text{FFNN}_a([\mathbf{g}_i, \mathbf{g}_j, \mathbf{g}_i \circ \mathbf{g}_j, \phi(i, j)])$$

where FFNN denotes a feed-forward neural network and $\circ$ denotes the element-wise product. θ_m, Θ_c and θ_a are network parameters. $\phi(i, j)$ is the feature vector encoding speaker and genre information from metadata.

The Reinforced Algorithm: We explore using the policy gradient algorithm to maximize the expected reward:

$$J(\theta) = \mathbb{E}_{a_{1:T} \sim p_\theta(a)} R(a_{1:T}) \quad (3)$$

Computing the exact gradient of $J(\theta)$ is infeasible due to the expectation over all possible action sequences. Instead, we use Monte-Carlo methods

Model	MUC			B^3			CEAFϕ_4			
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Avg. F1
Wiseman et al. (2016)	77.5	69.8	73.4	66.8	57.0	61.5	62.1	53.9	57.7	64.2
Clark and Manning (2016a)	79.2	70.4	74.6	69.9	58.0	63.4	63.5	55.5	59.2	65.7
Clark and Manning (2016b)	79.9	69.3	74.2	71.0	56.5	63.0	63.8	54.3	58.7	65.3
Lee et al. (2017)	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
Zhang et al. (2018)	79.4	73.8	76.5	69.0	62.3	65.5	64.9	58.3	61.4	67.8
Luan et al. (2018)*	78.6	77.1	77.9	66.3	65.4	65.9	66.0	63.1	64.5	69.4
Lee et al. (2018)*	81.4	**79.5**	80.4	72.2	**69.5**	70.8	68.2	**67.1**	67.6	73.0
Our base reinforced model	79.0	76.9	77.9	66.8	64.9	65.8	66.5	63.0	64.7	69.5
+ Entropy Regularization	79.6	77.2	78.4	70.7	65.1	67.8	67.6	63.4	65.4	70.5
+ ELMo embedding*	**85.4**	77.9	**81.4**	**77.9**	66.4	**71.7**	**70.6**	66.3	**68.4**	**73.8**

Table 1: Experimental results with MUC, B^3 and CEAFϕ_4 metrics on the test set of English OntoNotes. The models marked with * utilized word embedding from deep language model ELMo (Peters et al., 2018). The F1 improvement is statistically significant under t-test with $p < 0.05$, compared with Lee et al. (2018).

to approximate the actual gradient by randomly sampling N_s trajectories according to p_θ and compute the gradient only over the sampled trajectories. Meanwhile, following Clark and Manning (2016a), we subtract a baseline value from the reward to reduce the variance of gradient estimation. The gradient estimate is as follows:

$$\nabla_\theta J(\theta) \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \sum_{t=1}^{T} \nabla_\theta \log p_\theta(a_{it}|S_{it})(R_{\tau_i} - b)$$

where N_s is the number of sampled trajectories, $\tau_i = \{a_{i1}, \cdots a_{iT}\}$ is the ith sampled trajectory and $b = \sum_{i=1}^{N_s} R(\tau_i)/N_s$ is the baseline reward.

The Entropy Regularization: To prevent our model from being stuck in highly-peaked polices towards a few actions, an entropy regularization term is added to encourage exploration. The final regularized policy gradient estimate is as follows:

$$\nabla_\theta J(\theta) \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \sum_{t=1}^{T} \nabla_\theta \big[\log p_\theta(a_{it}|S_{it}) + \lambda_{expr} \\ p_\theta(a_{it}|S_{it}) \log p_\theta(a_{it}|S_{it}) \big] (R_{\tau_i} - b)$$

where $\lambda_{expr} \geq 0$ is the regularization parameter that controls how diverse our model can explore. The larger the λ_{expr} is, the more diverse our model can explore. If $\lambda_{expr} \to \infty$, all actions will be sampled uniformly regardless of current policies. To the contrary, if $\lambda_{expr} = 0$, all actions will be sampled based on current polices.

Pretraining: We pretrain the policy network parameterized by θ using the loss function below:

$$L(\theta) = -\sum_{i=1}^{N} \sum_{j \in \mathcal{Y}_i} I(i,j) \log \left(p(j|i;\theta)\right) \quad (4)$$

where N is the number of mentions, $I(i,j) = 1$ if mention i and j are coreferred, and 0 otherwise. $\mathcal{Y}_i$ is the set of candidate antecedents of mention i.

4 Experiments

We evaluate our model on the English OntoNotes v5.0 (Pradhan et al., 2011), which contains 2,802 training documents, 343 development documents, and 348 test documents. We reuse the hyperparameters and evaluation metrics from Lee et al. (2018) with a few exceptions. First, we pretrain our model using Eq. (4) for around 200K steps and use the learned parameters for initialization. Besides, we set the number of sampled trajectories $N_s = 100$, tune the regularization parameter λ_{expr} in $\{10^{-5}, 10^{-4}, 0.001, 0.01, 0.1, 1\}$ and set it to 10^{-4} based on the development set.

We use three standard metrics: MUC (Grishman and Sundheim, 1995), B^3 (Recasens and Hovy, 2011) and CEAFϕ_4 (Cai and Strube, 2010). For each metric, we report the precision, recall and F1 score. The final evaluation is the average F1 of the above three metrics.

4.1 Results

In Table 1, we compare our model with the coreference systems that have produced significant improvement over the last 3 years on the OntoNotes benchmark. The reported results are either adopted from their papers or reproduced from their code. The first section of the table lists the pipeline models, while the second section lists the end-to-end approaches. The third section lists the results of our model with different variants. Note that Luan et al. (2018)'s method contains 3 tasks: named entity recognition, relation inference and coreference resolution and we disable the relation inference task and train the other two tasks.

Built on top of the model in Lee et al. (2018) but excluding ELMo, our base reinforced model improves the average F1 score around 2 points (statistical significant t-test with $p < 0.05$) compared

with Lee et al. (2017); Zhang et al. (2018). Besides, it is even comparable with the end-to-end multi-task coreference model that has ELMo support (Luan et al., 2018), which demonstrates the power of reinforcement learning combined with the state-of-the-art end-to-end model in Lee et al. (2018). Regarding our model, using entropy regularization to encourage exploration can improve the result by 1 point. Moreover, introducing the context-dependent ELMo embedding to our base model can further boosts the performance, which is consistent with the results in Lee et al. (2018). We also notice that our full model's improvement is mainly from higher precision scores and reasonably good recall scores, which indicates that our reinforced model combined with more active exploration produces better coreference scores to reduce false positive coreference links.

Overall, our full model achieves the state-of-the-art performance of 73.8% F1-score when using ELMo and entropy regularization (compared to models marked with * in Table 1), and our approach simultaneously obtains the best F1-score of 70.5% when using fixed word embedding *only*.

Model	Prec.	Rec.	F1
Our full model	**89.6**	82.2	**85.7**
Lee et al. (2018)	86.2	**83.7**	84.9

Table 2: The overall mention detection results on the test set of OntoNotes. The F1 improvement is statistically significant under t-test with $p < 0.05$.

Since mention detection is a subtask of coreference resolution, it is worthwhile to study the performance. Table 2 shows the mention detection results on the test set. Similar to coreference linking results, our model achieves higher precision and F1 score, which indicates that our model can significantly reduce false positive mentions while it can still find a reasonable number of mentions.

4.2 Analysis and Discussion

Ablation Study: To understand the effect of different components, we conduct an ablation study on the development set as illustrated in Table 3. Clearly, removing entropy regularization deteriorates the average F1 score by 1%. Also, disabling coarse-to-fine pruning or second-order inference decreases 0.3/0.5 F1 score. Among all the components, ELMo embedding makes the most contribution and improves the result by 3.1%.

Model	Avg. F1
Full Model	74.1
w/o entropy regularization	73.1
w/o coarse-to-fine pruning	73.8
w/o second-order inference	73.6
w/o ELMo embedding	71.0

Table 3: Ablation study on the development set. "Coarse-to-fine pruning" and "second-order inference" are adopted from Lee et al. (2018)

Impact of the parameter λ_{expr}: Since the parameter λ_{expr} directly controls how diverse the model is explored during training, it is necessary to study its effect on the model performance. Figure 4 shows the avg. F1 score on the development set for our full model and Lee et al. (2018). We observe that λ_{expr} does have a strong effect on the performance and the best value is around 10^{-4}. Besides, our full model consistently outperforms Lee et al. (2018) over a wide range of λ_{expr}.

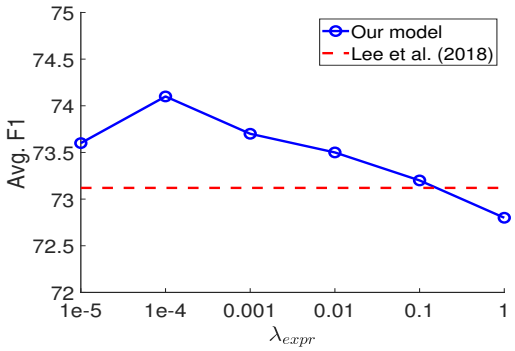

Figure 4: Avg. F1 score on the development set with different regularization parameter λ_{expr}. The result of Lee et al. (2018) is also plotted for comparison, which is a flat line since it does not depend on λ_{expr}.

5 Conclusion

We present the first end-to-end reinforcement learning based coreference resolution model. Our model transforms the supervised higher order coreference model to a policy gradient model that can directly optimizes coreference evaluation metrics. Experiments on the English OntoNotes benchmark demonstrate that our full model integrated with entropy regularization significantly outperforms previous coreference systems.

There are several potential improvements to our model as future work, such as incorporating mention detection result as a part of the reward. Another interesting direction would be introducing intermediate step rewards for each action to better guide the behaviour of the RL agent.

References

Jie Cai and Michael Strube. 2010. Evaluation metrics for end-to-end coreference resolution systems. In *Proceedings the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 28–36, Tokyo, Japan.

Kevin Clark and Christopher D. Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2256–2262, Austin, TX.

Kevin Clark and Christopher D Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 643–653, Berlin, Germany.

Zeyu Dai, Hongliang Fei, and Ping Li. 2019. Coreference aware representation learning for neural named entity recognition. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Macau.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. 2019. Diversity is all you need: Learning skills without a reward function. In *Seventh International Conference on Learning Representations (ICLR)*, New Orleans, LA.

Ralph Grishman and Beth Sundheim. 1995. Design of the muc-6 evaluation. In *Proceedings of the 6th conference on Message understanding (MUC)*, pages 1–11, Columbia, MD.

Aria Haghighi and Dan Klein. 2010. Coreference resolution in a modular, entity-centered model. In *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics (NAACL)*, pages 385–393, Los Angeles, CA.

Hannaneh Hajishirzi, Leila Zilles, Daniel S. Weld, and Luke S. Zettlemoyer. 2013. Joint coreference resolution and named-entity linking with multi-pass sieves. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 289–299, Seattle, WA.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 188–197, Copenhagen, Denmark.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 687–692, New Orleans, LA.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3219–3232, Brussels, Belgium.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, New York, NY.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 2227–2237, New Orleans, LA.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. Conll-2011 shared task: Modeling unrestricted coreference in ontonotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27, Portland, OR.

Marta Recasens and Eduard Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 17(4):485–510.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 994–1004, San Diego, CA.

Qingyu Yin, Yu Zhang, Weinan Zhang, Ting Liu, and William Yang Wang. 2018. Deep reinforcement learning for chinese zero pronoun resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 569–578, Melbourne, Australia.

Rui Zhang, Cícero Nogueira dos Santos, Michihiro Yasunaga, Bing Xiang, and Dragomir R. Radev. 2018. Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 102–107, Melbourne, Australia.