Higher-order Coreference Resolution with Coarse-to-fine Inference

Kenton Lee **Luheng He** **Luke Zettlemoyer**
Paul G. Allen School of Computer Science & Engineering
University of Washington, Seattle WA
{kentonl, luheng, lsz}@cs.washington.edu

Abstract

We introduce a fully differentiable approximation to higher-order inference for coreference resolution. Our approach uses the antecedent distribution from a span-ranking architecture as an attention mechanism to iteratively refine span representations. This enables the model to softly consider multiple hops in the predicted clusters. To alleviate the computational cost of this iterative process, we introduce a coarse-to-fine approach that incorporates a less accurate but more efficient bilinear factor, enabling more aggressive pruning without hurting accuracy. Compared to the existing state-of-the-art span-ranking approach, our model significantly improves accuracy on the English OntoNotes benchmark, while being far more computationally efficient.

1 Introduction

Recent coreference resolution systems have heavily relied on first order models (Clark and Manning, 2016a; Lee et al., 2017), where only pairs of entity mentions are scored by the model. These models are computationally efficient and scalable to long documents. However, because they make independent decisions about coreference links, they are susceptible to predicting clusters that are locally consistent but globally inconsistent. Figure 1 shows an example from Wiseman et al. (2016) that illustrates this failure case. The plurality of **[you]** is underspecified, making it locally compatible with both **[I]** and **[all of you]**, while the full cluster would have mixed plurality, resulting in global inconsistency.

Speaker 1: Um and **[I]** think that is what's - Go ahead Linda.
Speaker 2: Well and uh thanks goes to **[you]** and to the media to help us... So our hat is off to **[all of you]** as well.

Figure 1: Example of consistency errors to which first-order span-ranking models are susceptible. Span pairs (**I**, **you**) and (**you**, **all of you**) are locally consistent, but the span triplet (**I**, **you**, **all of you**) is globally inconsistent. Avoiding this error requires modeling higher-order structures.

We introduce an approximation of higher-order inference that uses the span-ranking architecture from Lee et al. (2017) in an iterative manner. At each iteration, the antecedent distribution is used as an attention mechanism to optionally update existing span representations, enabling later coreference decisions to softly condition on earlier coreference decisions. For the example in Figure 1, this enables the linking of **[you]** and **[all of you]** to depend on the linking of **[I]** and **[you]**.

To alleviate computational challenges from this higher-order inference, we also propose a coarse-to-fine approach that is learned with a single end-to-end objective. We introduce a less accurate but more efficient coarse factor in the pairwise scoring function. This additional factor enables an extra pruning step during inference that reduces the number of antecedents considered by the more accurate but inefficient fine factor. Intuitively, the model cheaply computes a rough sketch of *likely* antecedents before applying a more expensive scoring function.

Our experiments show that both of the above contributions improve the performance of coreference resolution on the English OntoNotes benchmark. We observe a significant increase in average F1 with a second-order model, but returns quickly diminish with a third-order model. Additionally, our analysis shows that the coarse-to-fine approach makes the model performance relatively insensitive to more aggressive antecedent pruning, compared to the distance-based heuristic pruning from previous work.

Proceedings of NAACL-HLT 2018, pages 687–692
New Orleans, Louisiana, June 1 - 6, 2018. ©2018 Association for Computational Linguistics

2 Background

Task definition We formulate the coreference resolution task as a set of antecedent assignments y_i for each of span i in the given document, following Lee et al. (2017). The set of possible assignments for each y_i is $\mathcal{Y}(i) = \{\epsilon, 1, \ldots, i-1\}$, a dummy antecedent ϵ and all preceding spans. Non-dummy antecedents represent coreference links between i and y_i. The dummy antecedent ϵ represents two possible scenarios: (1) the span is not an entity mention or (2) the span is an entity mention but it is not coreferent with any previous span. These decisions implicitly define a final clustering, which can be recovered by grouping together all spans that are connected by the set of antecedent predictions.

Baseline We describe the baseline model (Lee et al., 2017), which we will improve to address the modeling and computational limitations discussed previously. The goal is to learn a distribution $P(y_i)$ over antecedents for each span i :

$$P(y_i) = \frac{e^{s(i, y_i)}}{\sum_{y' \in \mathcal{Y}(i)} e^{s(i, y')}} \tag{1}$$

where $s(i, j)$ is a pairwise score for a coreference link between span i and span j. The baseline model includes three factors for this pairwise coreference score: (1) $s_{\mathrm{m}}(i)$, whether span i is a mention, (2) $s_{\mathrm{m}}(j)$, whether span j is a mention, and (3) $s_{\mathrm{a}}(i, j)$ whether j is an antecedent of i:

$$s(i, j) = s_{\mathrm{m}}(i) + s_{\mathrm{m}}(j) + s_{\mathrm{a}}(i, j) \tag{2}$$

In the special case of the dummy antecedent, the score $s(i, \epsilon)$ is instead fixed to 0. A common component used throughout the model is the vector representations $\boldsymbol{g}_i$ for each possible span i. These are computed via bidirectional LSTMs (Hochreiter and Schmidhuber, 1997) that learn context-dependent boundary and head representations. The scoring functions s_{m} and s_{a} take these span representations as input:

$$s_{\mathrm{m}}(i) = \boldsymbol{w}_{\mathrm{m}}^{\top} \mathrm{FFNN}_{\mathrm{m}}(\boldsymbol{g}_i) \tag{3}$$

$$s_{\mathrm{a}}(i, j) = \boldsymbol{w}_{\mathrm{a}}^{\top} \mathrm{FFNN}_{\mathrm{a}}([\boldsymbol{g}_i, \boldsymbol{g}_j, \boldsymbol{g}_i \circ \boldsymbol{g}_j, \phi(i, j)]) \tag{4}$$

where $\circ$ denotes element-wise multiplication, FFNN denotes a feed-forward neural network, and the antecedent scoring function $s_{\mathrm{a}}(i, j)$ includes explicit element-wise similarity of each span $\boldsymbol{g}_i \circ \boldsymbol{g}_j$ and a feature vector $\phi(i, j)$ encoding speaker and genre information from the metadata and the distance between the two spans.

The model above is factored to enable a two-stage beam search. A beam of up to M potential mentions is computed (where M is proportional to the document length) based on the spans with the highest mention scores $s_{\mathrm{m}}(i)$. Pairwise coreference scores are only computed between surviving mentions during both training and inference.

Given supervision of gold coreference clusters, the model is learned by optimizing the marginal log-likelihood of the possibly correct antecedents. This marginalization is required since the best antecedent for each span is a latent variable.

3 Higher-order Coreference Resolution

The baseline above is a first-order model, since it only considers pairs of spans. First-order models are susceptible to consistency errors as demonstrated in Figure 1. Unlike in sentence-level semantics, where higher-order decisions can be implicitly modeled by the LSTMs, modeling these decisions at the document-level requires explicit inference due to the potentially very large surface distance between mentions.

We propose an inference procedure that allows the model to condition on higher-order structures, while being fully differentiable. This inference involves N iterations of refining span representations, denoted as $\boldsymbol{g}_i^n$ for the representation of span i at iteration n. At iteration n, $\boldsymbol{g}_i^n$ is computed with an attention mechanism that averages over previous representations $\boldsymbol{g}_j^{n-1}$ weighted according to how likely each mention j is to be an antecedent for i, as defined below.

The baseline model is used to initialize the span representation at $\boldsymbol{g}_i^1$. The refined span representations allow the model to also iteratively refine the antecedent distributions $P_n(y_i)$:

$$P_n(y_i) = \frac{e^{s(\boldsymbol{g}_i^n, \boldsymbol{g}_{y_i}^n)}}{\sum_{y \in \mathcal{Y}(i)} e^{s(\boldsymbol{g}_i^n, \boldsymbol{g}_y^n))}} \tag{5}$$

where s is the coreference scoring function of the baseline architecture. The scoring function uses the same parameters at every iteration, but it is given different span representations.

At each iteration, we first compute the expected antecedent representation $\boldsymbol{a}_i^n$ of each span i by using the current antecedent distribution $P_n(y_i)$ as an attention mechanism:

$$\boldsymbol{a}_i^n = \sum_{y_i \in \mathcal{Y}(i)} P_n(y_i) \cdot \boldsymbol{g}_{y_i}^n \tag{6}$$

The current span representation $\boldsymbol{g}_i^n$ is then updated via interpolation with its expected antecedent representation $\boldsymbol{a}_i^n$:

$$\boldsymbol{f}_i^n = \sigma(\mathbf{W}_{\mathrm{f}}[\boldsymbol{g}_i^n, \boldsymbol{a}_i^n]) \tag{7}$$

$$\boldsymbol{g}_i^{n+1} = \boldsymbol{f}_i^n \circ \boldsymbol{g}_i^n + (\mathbf{1} - \boldsymbol{f}_i^n) \circ \boldsymbol{a}_i^n \tag{8}$$

The learned gate vector $\boldsymbol{f}_i^n$ determines for each dimension whether to keep the current span information or to integrate new information from its expected antecedent. At iteration n, $\boldsymbol{g}_i^n$ is an element-wise weighted average of approximately n span representations (assuming $P_n(y_i)$ is peaked), allowing $P_n(y_i)$ to softly condition on up to n other spans in the predicted cluster.

Span-ranking can be viewed as predicting latent antecedent trees (Fernandes et al., 2012; Martschat and Strube, 2015), where the predicted antecedent is the parent of a span and each tree is a predicted cluster. By iteratively refining the span representations and antecedent distributions, another way to interpret this model is that the joint distribution $\prod_i P_N(y_i)$ implicitly models every directed path of up to length $N + 1$ in the latent antecedent tree.

4 Coarse-to-fine Inference

The model described above scales poorly to long documents. Despite heavy pruning of potential mentions, the space of possible antecedents for every surviving span is still too large to fully consider. The bottleneck is in the antecedent score $s_{\mathrm{a}}(i, j)$, which requires computing a tensor of size $M \times M \times (3|\boldsymbol{g}| + |\phi|)$.

This computational challenge is even more problematic with the iterative inference from Section 3, which requires recomputing this tensor at every iteration.

4.1 Heuristic antecedent pruning

To reduce computation, Lee et al. (2017) heuristically consider only the nearest K antecedents of each span, resulting in a smaller input of size $M \times K \times (3|\boldsymbol{g}| + |\phi|)$.

The main drawback to this solution is that it imposes an a priori limit on the maximum distance of a coreference link. The previous work only considers up to $K = 250$ nearest mentions, whereas coreference links can reach much further in natural language discourse.

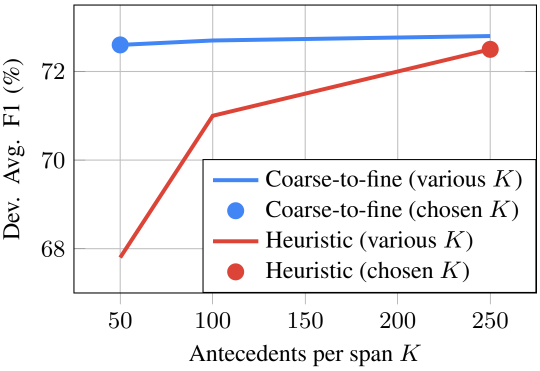

Figure 2: Comparison of accuracy on the development set for the two antecedent pruning strategies with various beams sizes K. The distance-based heuristic pruning performance drops by almost 5 F1 when reducing K from 250 to 50, while the coarse-to-fine pruning results in an insignificant drop of less than 0.2 F1.

4.2 Coarse-to-fine antecedent pruning

We instead propose a coarse-to-fine approach that can be learned end-to-end and does not establish an a priori maximum coreference distance. The key component of this coarse-to-fine approach is an alternate bilinear scoring function:

$$s_{\mathrm{c}}(i, j) = \boldsymbol{g}_i^\top \mathbf{W}_{\mathrm{c}}\, \boldsymbol{g}_j \tag{9}$$

where $\mathbf{W}_{\mathrm{c}}$ is a learned weight matrix. In contrast to the concatenation-based $s_{\mathrm{a}}(i, j)$, the bilinear $s_{\mathrm{c}}(i, j)$ is far less accurate. A direct replacement of $s_{\mathrm{a}}(i, j)$ with $s_{\mathrm{c}}(i, j)$ results in a performance loss of over 3 F1 in our experiments. However, $s_{\mathrm{c}}(i, j)$ is much more efficient to compute. Computing $s_{\mathrm{c}}(i, j)$ only requires manipulating matrices of size $M \times |\boldsymbol{g}|$ and $M \times M$.

Therefore, we instead propose to use $s_{\mathrm{c}}(i, j)$ to compute a rough sketch of *likely* antecedents. This is accomplished by including it as an additional factor in the model:

$$s(i, j) = s_{\mathrm{m}}(i) + s_{\mathrm{m}}(j) + s_{\mathrm{c}}(i, j) + s_{\mathrm{a}}(i, j) \tag{10}$$

Similar to the baseline model, we leverage this additional factor to perform an additional beam pruning step. The final inference procedure involves a three-stage beam search:

First stage Keep the top M spans based on the mention score $s_{\mathrm{m}}(i)$ of each span.

Second stage Keep the top K antecedents of each remaining span i based on the first three factors, $s_{\mathrm{m}}(i) + s_{\mathrm{m}}(j) + s_{\mathrm{c}}(i, j)$.

	MUC			B^3			$CEAF_{\phi_4}$			
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Avg. F1
Martschat and Strube (2015)	76.7	68.1	72.2	66.1	54.2	59.6	59.5	52.3	55.7	62.5
Clark and Manning (2015)	76.1	69.4	72.6	65.6	56.0	60.4	59.4	53.0	56.0	63.0
Wiseman et al. (2015)	76.2	69.3	72.6	66.2	55.8	60.5	59.4	54.9	57.1	63.4
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 (2016b)	79.9	69.3	74.2	71.0	56.5	63.0	63.8	54.3	58.7	65.3
Clark and Manning (2016a)	79.2	70.4	74.6	69.9	58.0	63.4	63.5	55.5	59.2	65.7
Lee et al. (2017)	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
+ `ELMo` (Peters et al., 2018)	80.1	77.2	78.6	69.8	66.5	68.1	66.4	62.9	64.6	70.4
+ hyperparameter tuning	80.7	78.8	79.8	71.7	68.7	70.2	67.2	66.8	67.0	72.3
+ coarse-to-fine inference	80.4	**79.9**	80.1	71.0	**70.0**	70.5	67.5	**67.2**	67.3	72.6
+ second-order inference	**81.4**	79.5	**80.4**	**72.2**	69.5	**70.8**	**68.2**	67.1	**67.6**	**73.0**

Table 1: Results on the test set on the English CoNLL-2012 shared task. The average F1 of MUC, B^3, and $CEAF_{\phi_4}$ is the main evaluation metric. We show only non-ensembled models for fair comparison.

Third stage The overall coreference $s(i, j)$ is computed based on the remaining span pairs. The soft higher-order inference from Section 3 is computed in this final stage.

While the maximum-likelihood objective is computed over only the span pairs from this final stage, this coarse-to-fine approach expands the set of coreference links that the model is capable of learning. It achieves better performance while using a much smaller K (see Figure 2).

5 Experimental Setup

We use the English coreference resolution data from the CoNLL-2012 shared task (Pradhan et al., 2012) in our experiments. The code for replicating these results is publicly available.[1]

Our models reuse the hyperparameters from Lee et al. (2017), with a few exceptions mentioned below. In our results, we report two improvements that are orthogonal to our contributions.

- We used embedding representations from a language model (Peters et al., 2018) at the input to the LSTMs (`ELMo` in the results).

- We changed several hyperparameters:

    1. increasing the maximum span width from 10 to 30 words.
    2. using 3 highway LSTMs instead of 1.
    3. using GloVe word embeddings (Pennington et al., 2014) with a window size of 2 for the head word embeddings and a window size of 10 for the LSTM inputs.

The baseline model considers up to 250 antecedents per span. As shown in Figure 2, the coarse-to-fine model is quite insensitive to more aggressive pruning. Therefore, our final model considers only 50 antecedents per span.

On the development set, the second-order model ($N = 2$) outperforms the first-order model by 0.8 F1, but the third order model only provides an additional 0.1 F1 improvement. Therefore, we only compute test results for the second-order model.

6 Results

We report the precision, recall, and F1 of the the MUC, B^3, and $CEAF_{\phi_4}$ metrics using the official CoNLL-2012 evaluation scripts. The main evaluation is the average F1 of the three metrics.

Results on the test set are shown in Table 1. We include performance of systems proposed in the past 3 years for reference. The baseline relative to our contributions is the span-ranking model from Lee et al. (2017) augmented with both `ELMo` and hyperparameter tuning, which achieves 72.3 F1. Our full approach achieves 73.0 F1, setting a new state of the art for coreference resolution.

Compared to the heuristic pruning with up to 250 antecedents, our coarse-to-fine model only computes the expensive scores $s_a(i, j)$ for 50 antecedents. Despite using far less computation, it outperforms the baseline because the coarse scores

[1] `https://github.com/kentonl/e2e-coref`

$s_c(i, j)$ can be computed for all antecedents, enabling the model to potentially predict a coreference link between any two spans in the document. As a result, we observe a much higher recall when adopting the coarse-to-fine approach.

We also observe further improvement by including the second-order inference (Section 3). The improvement is largely driven by the overall increase in precision, which is expected since the higher-order inference mainly serves to rule out inconsistent clusters. It is also consistent with findings from Martschat and Strube (2015) who report mainly improvements in precision when modeling latent trees to achieve a similar goal.

7 Related Work

In addition to the end-to-end span-ranking model (Lee et al., 2017) that our proposed model builds upon, there is a large body of literature on coreference resolvers that fundamentally rely on scoring span pairs (Ng and Cardie, 2002; Bengtson and Roth, 2008; Denis and Baldridge, 2008; Fernandes et al., 2012; Durrett and Klein, 2013; Wiseman et al., 2015; Clark and Manning, 2016a).

Motivated by structural consistency issues discussed above, significant effort has also been devoted towards cluster-level modeling. Since global features are notoriously difficult to define (Wiseman et al., 2016), they often depend heavily on existing pairwise features or architectures (Björkelund and Kuhn, 2014; Clark and Manning, 2015, 2016b). We similarly use an existing pairwise span-ranking architecture as a building block for modeling more complex structures. In contrast to Wiseman et al. (2016) who use highly expressive recurrent neural networks to model clusters, we show that the addition of a relatively lightweight gating mechanism is sufficient to effectively model higher-order structures.

8 Conclusion

We presented a state-of-the-art coreference resolution system that models higher order interactions between spans in predicted clusters. Additionally, our proposed coarse-to-fine approach alleviates the additional computational cost of higher-order inference, while maintaining the end-to-end learnability of the entire model.

Acknowledgements

The research was supported in part by DARPA under the DEFT program (FA8750-13-2-0019), the ARO (W911NF-16-1-0121), the NSF (IIS-1252835, IIS-1562364), gifts from Google and Tencent, and an Allen Distinguished Investigator Award. We also thank the UW NLP group for helpful conversations and comments on the work.

References

Eric Bengtson and Dan Roth. 2008. Understanding the value of features for coreference resolution. In *EMNLP*.

Anders Björkelund and Jonas Kuhn. 2014. Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *ACL*.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. In *ACL*.

Kevin Clark and Christopher D. Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. In *EMNLP*.

Kevin Clark and Christopher D. Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. In *ACL*.

Pascal Denis and Jason Baldridge. 2008. Specialized models and ranking for coreference resolution. In *EMNLP*.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *EMNLP*.

Eraldo Rezende Fernandes, Cícero Nogueira Dos Santos, and Ruy Luiz Milidiú. 2012. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *CoNLL*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural computation* .

Kenton Lee, Luheng He, Mike Lewis, and Luke S. Zettlemoyer. 2017. End-to-end neural coreference resolution. In *EMNLP*.

Sebastian Martschat and Michael Strube. 2015. Latent structures for coreference resolution. *TACL* .

Vincent Ng and Claire Cardie. 2002. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. *Computational linguistics* .

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *HLT-NAACL*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *CoNLL*.

Sam Wiseman, Alexander M Rush, and Stuart M Shieber. 2016. Learning global features for coreference resolution. In *NAACL-HLT*.

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