Neural Coreference Resolution with Deep Biaffine Attention by Joint Mention Detection and Mention Clustering

Rui Zhang *
Yale University
r.zhang@yale.edu

Cícero Nogueira dos Santos
IBM Research
cicerons@us.ibm.com

Michihiro Yasunaga
Yale University
michihiro.yasunaga@yale.edu

Bing Xiang
IBM Watson
bingxia@us.ibm.com

Dragomir R. Radev
Yale University
dragomir.radev@yale.edu

Abstract

Coreference resolution aims to identify in a text all mentions that refer to the same real-world entity. The state-of-the-art end-to-end neural coreference model considers all text spans in a document as potential mentions and learns to link an antecedent for each possible mention. In this paper, we propose to improve the end-to-end coreference resolution system by (1) using a biaffine attention model to get antecedent scores for each possible mention, and (2) jointly optimizing the mention detection accuracy and the mention clustering log-likelihood given the mention cluster labels. Our model achieves the state-of-the-art performance on the CoNLL-2012 Shared Task English test set.

1 Introduction

End-to-end coreference resolution is the task of identifying and grouping *mentions* in a text such that all mentions in a cluster refer to the same entity. An example is given below (Björkelund and Kuhn, 2014) where mentions for two entities are labeled in two clusters:

> [Drug Emporium Inc.]$_{a1}$ said [Gary Wilber]$_{b1}$ was named CEO of [this drugstore chain]$_{a2}$. [He]$_{b2}$ succeeds his father, Philip T. Wilber, who founded [the company]$_{a3}$ and remains chairman. Robert E. Lyons III, who headed the [company]$_{a4}$'s Philadelphia region, was appointed president and chief operating officer, succeeding [Gary Wilber]$_{b3}$.

Many traditional coreference systems, either rule-based (Haghighi and Klein, 2009; Lee et al., 2011) or learning-based (Bengtson and Roth, 2008; Fernandes et al., 2012; Durrett and Klein, 2013; Björkelund and Kuhn, 2014), usually solve the problem in two separate stages: (1) a mention detector to propose entity mentions from the text, and (2) a coreference resolver to cluster proposed mentions. At both stages, they rely heavily on complicated, fine-grained, conjoined features via heuristics. This pipeline approach can cause cascading errors, and in addition, since both stages rely on a syntactic parser and complicated handcraft features, it is difficult to generalize to new data sets and languages.

Very recently, Lee et al. (2017) proposed the first state-of-the-art end-to-end neural coreference resolution system. They consider all text spans as potential mentions and therefore eliminate the need of carefully hand-engineered mention detection systems. In addition, thanks to the representation power of pre-trained word embeddings and deep neural networks, the model only uses a minimal set of hand-engineered features (speaker ID, document genre, span distance, span width).

The core of the end-to-end neural coreference resolver is the scoring function to compute the mention scores for all possible spans and the antecedent scores for a pair of spans. Furthermore, one major challenge of coreference resolution is that most mentions in the document are singleton or non-anaphoric, i.e., not coreferent with any previous mention (Wiseman et al., 2015). Since the data set only have annotations for mention clusters, the end-to-end coreference resolution system needs to detect mentions, detect anaphoricity, and perform coreference linking. Therefore, research questions still remain on good designs of the scoring architecture and the learning strategy for both mention detection and antecedent scoring given only the gold cluster labels.

To this end, we propose to use a biaffine atten-

*Work done during the internship at IBM Watson.

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 102–107
Melbourne, Australia, July 15 - 20, 2018.

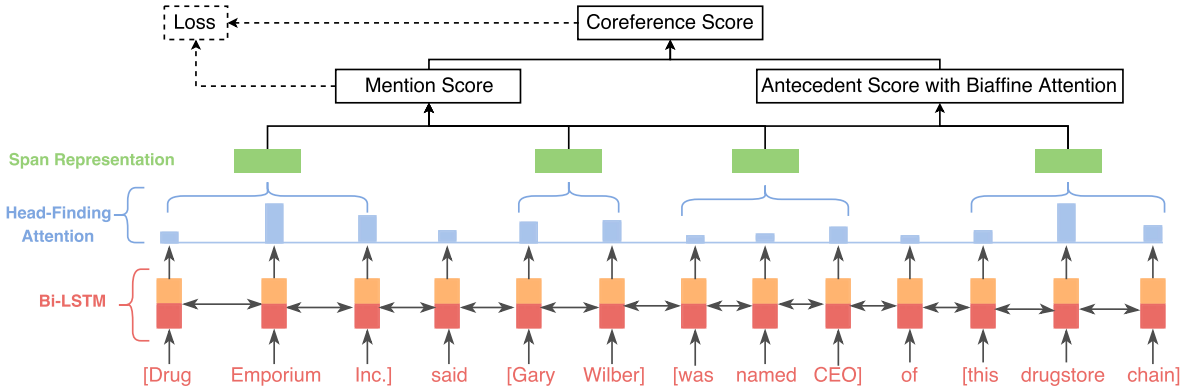

Figure 1: Model architecture. We consider all text spans up to 10-word length as possible mentions. For brevity, we only show three candidate antecedent spans ("Drug Emporium Inc.", "Gary Wilber", "was named CEO") for the current span "this drugstore chain".

tion model instead of pure feed forward networks to compute antecedent scores. Furthermore, instead of training only to maximize the marginal likelihood of gold antecedent spans, we jointly optimize the mention detection accuracy and the mention clustering log-likelihood given the mention cluster labels. We optimize mention detection loss explicitly to extract mentions and also perform anaphoricity detection.

We evaluate our model on the CoNLL-2012 English data set and achieve new state-of-the-art performances of 67.8% F1 score using a single model and 69.2% F1 score using a 5-model ensemble.

2 Task Formulation

In end-to-end coreference resolution, the input is a document D with T words, and the output is a set of mention clusters each of which refers to the same entity. A possible *span* is an N-gram within a single sentence. We consider all possible spans up to a predefined maximum width. To impose an ordering, spans are sorted by the start position $\text{START}(i)$ and then by the end position $\text{END}(i)$. For each span i the system needs to assign an antecedent a_i from all preceding spans or a dummy antecedent ϵ: $a_i \in \{\epsilon, 1, \ldots, i-1\}$. If a span j is a true antecedent of the span i, then we have $a_i = j$ and $1 \leq j \leq i-1$. The dummy antecedent ϵ represents two possibilities: (1) the span i is not an entity mention, or (2) the span i is an entity mention but not coreferent with any previous span. Finally, the system groups mentions according to coreference links to form the mention clusters.

3 Model

Figure 1 illustrates our model. We adopt the same span representation approach as in Lee et al. (2017) using bidirectional LSTMs and a head-finding attention. Thereafter, a feed forward network produces scores for spans being entity mentions. For antecedent scoring, we propose a biaffine attention model (Dozat and Manning, 2017) to produce distributions of possible antecedents. Our training data only provides gold mention cluster labels. To make best use of this information, we propose to jointly optimize the mention scoring and antecedent scoring in our loss function.

Span Representation Suppose the current sentence of length L is $[w_1, w_2, \ldots, w_L]$, we use $\mathbf{w}_t$ to denote the concatenation of fixed pretrained word embeddings and CNN character embeddings (dos Santos and Zadrozny, 2014) for word w_t. Bidirectional LSTMs (Hochreiter and Schmidhuber, 1997) recurrently encode each w_t:

$$\begin{aligned}\overrightarrow{\mathbf{h}}_t &= \text{LSTM}^{\text{forward}}(\overrightarrow{\mathbf{h}}_{t-1}, \mathbf{w}_t)\\ \overleftarrow{\mathbf{h}}_t &= \text{LSTM}^{\text{backward}}(\overleftarrow{\mathbf{h}}_{t+1}, \mathbf{w}_t)\\ \mathbf{h}_t &= [\overrightarrow{\mathbf{h}}_t, \overleftarrow{\mathbf{h}}_t]\end{aligned} \quad (1)$$

Then, the head-finding attention computes a score distribution over different words in a span s_i:

$$\begin{aligned}\alpha_t &= \mathbf{v}_\alpha^\intercal \text{FFNN}_\alpha(\mathbf{h}_t)\\ s_{i,t} &= \frac{\exp(\alpha_t)}{\sum_{k=\text{START}(i)}^{\text{END}(i)} \exp(\alpha_k)}\\ \mathbf{w}_i^{\text{head-att}} &= \sum_{t=\text{START}(i)}^{\text{END}(i)} s_{i,t}\mathbf{w}_t\end{aligned} \quad (2)$$

where FFNN is a feed forward network outputting a vector.

Effective span representations encode both contextual information and internal structure of spans. Therefore, we concatenate different vectors, including a feature vector $\phi(i)$ for the span size, to produce the span representation $\mathbf{s}_i$ for s_i:

$$\mathbf{s}_i = [\mathbf{h}_{\text{START}(i)}, \mathbf{h}_{\text{END}(i)}, \mathbf{w}_i^{\text{head-att}}, \phi(i)] \quad (3)$$

Mention Scoring The span representation is input to a feed forward network which measures if it is an entity mention using a score $m(i)$:

$$m(i) = \mathbf{v}_{\text{m}}^{\intercal}\text{FFNN}_{\text{m}}(\mathbf{s}_i) \quad (4)$$

Since we consider all possible spans, the number of spans is $O(T^2)$ and the number of span pairs is $O(T^4)$. Due to computation efficiency, we prune candidate spans during both inference and training. We keep λT spans with highest mention scores.

Biaffine Attention Antecedent Scoring Consider the current span s_i and its previous spans s_j ($1 \leq j \leq i-1$), we propose to use a biaffine attention model to produce scores $c(i,j)$:

$$\begin{aligned} \hat{\mathbf{s}}_{\text{i}} &= \text{FFNN}_{\text{anaphora}}(\mathbf{s}_{\text{i}}) \\ \hat{\mathbf{s}}_{\text{j}} &= \text{FFNN}_{\text{antecedent}}(\mathbf{s}_{\text{j}}), 1 \leq j \leq i-1 \\ c(i,j) &= \hat{\mathbf{s}}_{\text{j}}^{\intercal}\mathbf{U}_{\text{bi}}\hat{\mathbf{s}}_{\text{i}} + \mathbf{v}_{\text{bi}}^{\intercal}\hat{\mathbf{s}}_{\text{i}} \end{aligned} \quad (5)$$

$\text{FFNN}_{\text{anaphora}}$ and $\text{FFNN}_{\text{antecedent}}$ reduce span representation dimensions and only keep information relevant to coreference decisions. Compared with the traditional FFNN approach in Lee et al. (2017), biaffine attention directly models both the compatibility of s_i and s_j by $\hat{\mathbf{s}}_{\text{j}}^{\intercal}\mathbf{U}_{\text{bi}}\hat{\mathbf{s}}_{\text{i}}$ and the prior likelihood of s_i having an antecedent by $\mathbf{v}_{\text{bi}}^{\intercal}\hat{\mathbf{s}}_{\text{i}}$.

Inference The final coreference score $s(i,j)$ for span s_i and span s_j consists of three terms: (1) if s_i is a mention, (2) if s_j is a mention, (3) if s_j is an antecedent for s_i. Furthermore, for dummy antecedent ϵ, we fix the final score to be 0:

$$s(i,j) = \begin{cases} m(i) + m(j) + c(i,j), & j \neq \epsilon \\ 0, & j = \epsilon \end{cases} \quad (6)$$

During inference, the model only creates a link if the highest antecedent score is positive.

Joint Mention Detection and Mention Cluster During training, only mention cluster labels are available rather than antecedent links. Therefore, Lee et al. (2017) train the model end-to-end by maximizing the following marginal log-likelihood where $\text{GOLD}(i)$ are gold antecedents for s_i:

$$\mathcal{L}_{\text{cluster}}(i) = \log \frac{\sum_{j' \in \text{GOLD}(i)} \exp(s(i,j'))}{\sum_{j=\epsilon,0,\ldots,i-1} \exp(s(i,j))} \quad (7)$$

However, the initial pruning is completely random and the mention scoring model only receives distant supervision if we only optimize the above mention cluster performance. This makes learning slow and ineffective especially for mention detection. Based on this observation, we propose to directly optimize mention detection:

$$\mathcal{L}_{\text{detect}}(i) = y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \quad (8)$$

where $\hat{y}_i = \text{sigmoid}(m(i))$, $y_i = 1$ if and only if s_i is in one of the gold mention clusters. Our final loss combines mention detection and clustering:

$$\mathcal{L}_{\text{loss}} = -\lambda_{\text{detect}} \sum_{i=1}^{N} \mathcal{L}_{\text{detect}}(i) - \sum_{i'=1}^{N'} \mathcal{L}_{\text{cluster}}(i')$$

where N is the number of all possible spans, N' is the number of unpruned spans, and $\lambda_{\text{detection}}$ controls weights of two terms.

4 Experiments

Data Set and Evaluation We evaluate our model on the CoNLL-2012 Shared Task English data (Pradhan et al., 2012) which is based on the OntoNotes corpus (Hovy et al., 2006). It contains 2,802/343/348 train/development/test documents in different genres.

We use three standard metrics: MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and CEAF_{ϕ_4} (Luo, 2005). We report Precision, Recall, F1 for each metric and the average F1 as the final CoNLL score.

Implementation Details For fair comparisons, we follow the same hyperparameters as in Lee et al. (2017). We consider all spans up to 10 words and up to 250 antecedents. $\lambda = 0.4$ is used for span pruning. We use fixed concatenations of 300-dimension GloVe (Pennington et al., 2014) embeddings and 50-dimension embeddings from Turian et al. (2010). Character CNNs use 8-dimension learned embeddings and 50 kernels for each window size in {3,4,5}. LSTMs have hidden size 200, and each FFNN has two hidden layers with 150 units and ReLU (Nair and Hinton, 2010) activations. We include (speaker ID, document

	MUC			B^3			$CEAF_{\phi_4}$			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
Our work (5-model ensemble)	82.1	73.6	77.6	73.1	62.0	67.1	67.5	59.0	62.9	**69.2**
Lee et al. (2017) (5-model ensemble)	81.2	73.6	77.2	72.3	61.7	66.6	65.2	60.2	62.6	68.8
Our work (single model)	79.4	73.8	76.5	69.0	62.3	65.5	64.9	58.3	61.4	**67.8**
Lee et al. (2017) (single model)	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
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
Wiseman et al. (2016)	77.5	69.8	73.4	66.8	57.0	61.5	62.1	53.9	57.7	64.2
Wiseman et al. (2015)	76.2	69.3	72.6	66.2	55.8	60.5	59.4	54.9	57.1	63.4
Fernandes et al. (2014)	75.9	65.8	70.5	77.7	65.8	71.2	43.2	55.0	48.4	63.4
Clark and Manning (2015)	76.1	69.4	72.6	65.6	56.0	60.4	59.4	53.0	56.0	63.0
Martschat and Strube (2015)	76.7	68.1	72.2	66.1	54.2	59.6	59.5	52.3	55.7	62.5
Durrett and Klein (2014)	72.6	69.9	71.2	61.2	56.4	58.7	56.2	54.2	55.2	61.7
Björkelund and Kuhn (2014)	74.3	67.5	70.7	62.7	55.0	58.6	59.4	52.3	55.6	61.6
Durrett and Klein (2013)	72.9	65.9	69.2	63.6	52.5	57.5	54.3	54.4	54.3	60.3

Table 1: Experimental results on the CoNLL-2012 Englisth test set. The F1 improvements are statistical significant with $p < 0.05$ under the paired bootstrap resample test (Koehn, 2004) compared with Lee et al. (2017).

	Avg. F1
Our model (single)	67.8
without mention detection loss	67.5
without biaffine attention	67.4
Lee et al. (2017)	67.3

Table 2: Ablation study on the development set.

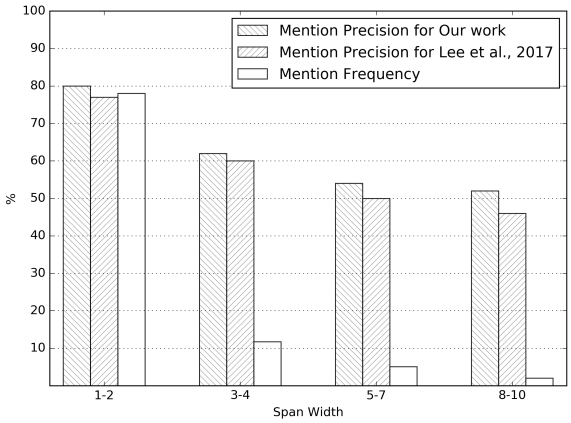

Figure 2: Mention detection subtask on development set. We plot accuracy and frequency breakdown by span widths.

genre, span distance, span width) features as 20-dimensional learned embeddings. Word and character embeddings use 0.5 dropout. All hidden layers and feature embeddings use 0.2 dropout. The batch size is 1 document. Based on the results on the development set, $\lambda_{\text{detection}} = 0.1$ works best from $\{0.05, 0.1, 0.5, 1.0\}$. Model is trained with ADAM optimizer (Kingma and Ba, 2015) and converges in around 200K updates, which is faster than that of Lee et al. (2017).

Overall Performance In Table 1, we compare our model with previous state-of-the-art systems. We obtain the best results in all F1 metrics. Our single model achieves 67.8% F1 and our 5-model ensemble achieves 69.2% F1. In particular, compared with Lee et al. (2017), our improvement mainly results from the precision scores. This indicates that the mention detection loss does produce better mention scores and the biaffine attention more effectively determines if two spans are coreferent.

Ablation Study To understand the effect of different proposed components, we perform ablation study on the development set. As shown in Table 2, removing the mention detection loss term or the biaffine attention decreases 0.3/0.4 final F1 score, but still higher than the baseline. This shows that both components have contributions and when they work together the total gain is even higher.

Mention Detection Subtask To further understand our model, we perform a mention detection subtask where spans with mention scores higher than 0 are considered as mentions. We show the mention detection accuracy breakdown by span widths in Figure 2. Our model indeed performs better thanks to the mention detection loss. The advantage is even clearer for longer spans which consist of 5 or more words.

In addition, it is important to note that our model can detect mentions that do not exist in the training data. While Moosavi and Strube (2017) observe that there is a large overlap between the gold mentions of the training and dev (test) sets, we find that our model can correctly de-

tect 1048 mentions which are not detected by Lee et al. (2017), consisting of 386 mentions existing in training data and 662 mentions not existing in training data. From those 662 mentions, some examples are (1) a suicide murder (2) Hong Kong Island (3) a US Airforce jet carrying robotic undersea vehicles (4) the investigation into who was behind the apparent suicide attack. This shows that our mention loss helps detection by generalizing to new mentions in test data rather than memorizing the existing mentions in training data.

5 Related Work

As summarized by Ng (2010), learning-based coreference models can be categorized into three types: (1) Mention-pair models train binary classifiers to determine if a pair of mentions are coreferent (Soon et al., 2001; Ng and Cardie, 2002; Bengtson and Roth, 2008). (2) Mention-ranking models explicitly rank all previous candidate mentions for the current mention and select a single highest scoring antecedent for each anaphoric mention (Denis and Baldridge, 2007b; Wiseman et al., 2015; Clark and Manning, 2016a; Lee et al., 2017). (3) Entity-mention models learn classifiers to determine whether the current mention is coreferent with a preceding, partially-formed mention cluster (Clark and Manning, 2015; Wiseman et al., 2016; Clark and Manning, 2016b).

In addition, we also note latent-antecedent models (Fernandes et al., 2012; Björkelund and Kuhn, 2014; Martschat and Strube, 2015). Fernandes et al. (2012) introduce coreference trees to represent mention clusters and learn to extract the maximum scoring tree in the graph of mentions.

Recently, several neural coreference resolution systems have achieved impressive gains (Wiseman et al., 2015, 2016; Clark and Manning, 2016b,a). They utilize distributed representations of mention pairs or mention clusters to dramatically reduce the number of hand-crafted features. For example, Wiseman et al. (2015) propose the first neural coreference resolution system by training a deep feed-forward neural network for mention ranking. However, these models still employ the two-stage pipeline and require a syntactic parser or a separate designed hand-engineered mention detector.

Finally, we also note the relevant work on joint mention detection and coreference resolution. Daumé III and Marcu (2005) propose to model both mention detection and coreference of the Entity Detection and Tracking task simultaneously. Denis and Baldridge (2007a) propose to use integer linear programming framework to model anaphoricity and coreference as a joint task.

6 Conclusion

In this paper, we propose to use a biaffine attention model to jointly optimize mention detection and mention clustering in the end-to-end neural coreference resolver. Our model achieves the state-of-the-art performance on the CoNLL-2012 Shared Task in English.

Acknowledgments

We thank Kenton Lee and three anonymous reviewers for their helpful discussion and feedback.

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*.

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*.

Hal Daumé III and Daniel Marcu. 2005. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*.

Pascal Denis and Jason Baldridge. 2007a. Joint determination of anaphoricity and coreference resolution using integer programming. In *NAACL*.

Pascal Denis and Jason Baldridge. 2007b. A ranking approach to pronoun resolution. In *IJCAI*.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. In *ICLR*.

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

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490.

Eraldo Rezende Fernandes, Cícero Nogueira Dos Santos, and Ruy Luiz Milidiú. 2012. Latent structure perceptron with feature induction for unrestricted coreference resolution. In *Joint Conference on EMNLP and CoNLL-Shared Task*.

Eraldo Rezende Fernandes, Cícero Nogueira dos Santos, and Ruy Luiz Milidiú. 2014. Latent trees for coreference resolution. *Computational Linguistics*.

Aria Haghighi and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *EMNLP*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% solution. In *NAACL*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP*.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the fifteenth conference on computational natural language learning: Shared task*, pages 28–34.

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

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *EMNLP*.

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

Nafise Sadat Moosavi and Michael Strube. 2017. Lexical features in coreference resolution: To be used with caution. In *ACL*.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*.

Vincent Ng. 2010. Supervised noun phrase coreference research: The first fifteen years. In *ACL*.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *ACL*.

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

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

Cícero Nogueira dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on Machine Learning (ICML), JMLR: W&CP volume 32*.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *ACL*.

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*.

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

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