A Study on Improving End-to-End Neural Coreference Resolution

Jia-Chen Gu, Zhen-Hua Ling, and Nitin Indurkhya

National Engineering Laboratory for Speech and Language Information Processing,
University of Science and Technology of China, Hefei, China
gujc@mail.ustc.edu.cn, {zhling, nitin}@ustc.edu.cn

Abstract. This paper studies the methods to improve end-to-end neural coreference resolution. First, we introduce a coreference cluster modification algorithm, which can help modify the coreference cluster to rule out the dissimilar mention in the cluster and reduce errors caused by the global inconsistence of coreference clusters. Additionally, we tune the model from two aspects to get more accurate coreference resolution results. On one hand, the simple scoring function is replaced with a feed-forward neural network when computing the head word scores for later attention mechanism which can help pick out the most important word. On the other hand, the maximum width of a mention is tuned. Our experimental results show that above methods improve the performance of coreference resolution effectively.

Keywords: Coreference resolution · End-to-end · Neural network

1 Introduction

Coreference resolution, the task of finding all expressions that refer to the same real-world entity in a text or dialogue, has become the core tasks of natural language processing (NLP) since the 1960s. An example of demonstrating the task of coreference resolution [1] that we need to resolve *I*, *my* and *she* as coreferential, *Nader* and *he* as coreferential respectively is showed in following text.

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

Nowadays, people are paying more and more attention to applying neural network to coreference resolution because neural-network-based models [2–5] have achieved impressive coreference resolution performance, especially the end-to-end neural model [5], which does not rely on syntactic parsers and many hand-engineered features. This end-to-end model makes independent decisions about whether two mentions are coreferential and then establish a coreference cluster through this kind of coreference relation. For example, if we make decisions that {*President of the People Republic of China*, *Xi Jinping*} and {*Xi Jinping*, *Mr.Xi*} are coreferential respectively, then we can get a cluster that {*President of the People Republic of China*, *Xi Jinping*, *Mr.Xi*} are coreferential naturally.

However, this model sometimes makes globally inconsistent decisions, and gets an incompletely correct cluster because of the independence between these decisions. To avoid this kind of error, a coreference cluster modification algorithm is proposed in this paper which can help rule out the mentions which are not globally coreferential within each cluster on the basis of the span-ranking architecture. After getting a coreference cluster through locally coreferential decisions, we use a scoring function to measure the extent of coreference relation between every mention pair. Then we establish a standard to decide whether to rule out the dissimilar mention in a coreference cluster.

Furthermore, we tune the hyperparameters from two aspects to get more accurate coreference resolution results. On one hand, to get more accurate scoring function to help measure the extent of coreference relation, we replace the scoring function with a feed-forward neural network when applying an attention mechanism [6] to compute the head word score. This modification enables the system to pick out the most important word more accurately to help express the representation of a mention which can help incorporate more information over words in a span. On the other hand, our experiments and analysis show that the model is susceptible to the maximum width of a mention, i.e. the number of words a mention can comprise of most. Therefore, we tune the maximum width of a mention in experiments.

Our experimental results show that the proposed coreference cluster modification algorithm can improve the performance of coreference resolution on the English OntoNotes benchmark. Our approach outperforms the baseline single model with an F1 improvement of 0.3. Additionally, we can also obtain an F1 improvement of 1.2 when tuning the hyperparameters of the model.

2 Related Work

Machine-learning-based methods for coreference resolution have developed for a long time since the first paper on machine-learning-based coreference resolution [7] was published. Hand-engineered systems built on top of automatically produced parse trees [8, 9] have achieved significant performance. Recently proposed neural-network-based models [2–4] outperformed all previous learning approaches. The more recent end-to-end neural model [5] has achieved further performance gains meanwhile it does not rely on syntactic parsers and hand-engineered features.

From a higher view of these approaches, all of the above models can be categorized as (1) mention-pair classifiers [10, 11], (2) entity-level models [4, 12, 13], (3) latent-tree models [14–16], (4) mention-ranking models [3, 9, 17], (5) span-ranking models [5]. Our proposed methods are based on the span-ranking model [5], which relies on scoring span pairs and then uses the scores to make coreference decisions. However, the end-to-end span-ranking model only concentrates on the direct link between span pairs while neglects the indirect link between the interval spans, which is the motivation of our proposed coreference cluster mofification algorithm.

3 Baseline Method

3.1 Task Definition

The end-to-end neural model [5] formulates the coreference resolution task as a set of antecedent assignments y_i for each of span i in the given document and our model follows the task formulation. The set of possible assignments for each y_i is $\mathcal{Y}(i)$ = {ϵ, 1, 2, ..., i - 1} which consists of 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 coreferential with any previous span. We can get a final clustering through these decisions, which may lead to the problem of global inconsistence of coreference cluster we have just mentioned above.

3.2 Baseline Model

The aim of the end-to-end baseline model [5] is to learn a distribution *P(y_i)* over antecedents for each span i as

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

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) $s_m(i)$, whether span i is a mention, (2) $s_m(j)$, whether span j is a mention, and (3) $s_a(i,\, j)$, whether span j is an antecedent of span i. $s(i,\, j)$ is calculated as

$$s(i,j) = s_m(i) + s_m(j) + s_a(i,j) \tag{2}$$

$s_m(i)$ and $s_a(i,\, j)$ are both functions of the span representation vector $\boldsymbol{g}_i$, which is computed via bidirectional LSTMs [18] and attention mechanism [6]. The detailed calculation of $s_m(i)$ and $s_a(i,\, j)$ are as follows

$$s_m(i) = \boldsymbol{w}_m^\top FFNN_m(\boldsymbol{g}_i) \tag{3}$$

$$s_a(i,j) = \boldsymbol{w}_a^\top FFNN_a([\boldsymbol{g}_i, \boldsymbol{g}_j, \boldsymbol{g}_i \circ \boldsymbol{g}_j, \phi(i,j)]) \tag{4}$$

where $\circ$ denotes element-wise multiplication, FFNN denotes a feed-forward neural network, and the antecedent scoring function $s_a(i,\, j)$ 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 span representation $\boldsymbol{g}_i$ is composed of boundary representation, head word vector and feature vector. We will restrict our discussion to the head word vector. The baseline model learns a task-specific notion of headness using an attention mechanism [6] over words in each span:

$$\alpha_t = \boldsymbol{w}_\alpha \cdot Projection(\boldsymbol{x}_t^*) \tag{5}$$

$$a_{i,t} = \frac{exp(\alpha_t)}{\sum_{k=START(i)}^{END(i)} exp(\alpha_k)} \tag{6}$$

$$\hat{\boldsymbol{x}}_i = \sum_{t=START(i)}^{END(i)} a_{i,t} \cdot \boldsymbol{x}_t \tag{7}$$

where $\hat{\boldsymbol{x}}_i$ is a weighted sum of word vectors in span i.

Given supervision of gold coreference clusters, the model is learned by optimizing the marginal log-likelihood of the possibly correct antecedents [5]. This marginalization is required since the best antecedent for each span is a latent variable.

3.3 Clustering Rules

The baseline model makes decisions about whether span i and span j are coreferential while these decisions are independent between each other. The baseline model obeys the following rules to make clustering decisions as follows.

- Span i has a set of scores, i.e. $s_a(i, j)$ with its every candidate antecedent in its antecedents set $\mathcal{Y}(i)$ = {ϵ, 1, 2, ..., i - 1} which can measure the extent of coreference relation.
- The span i picks out the one which has the highest score to be its antecedent and establishes a coreference link with its antecedent span.
- If span pairs {span i, span j} and {span j, span k} are both linked, span pair {span i, span k} will be linked naturally.

It is noticeable that span pair {span i, span k} will be linked naturally without confirming whether these two spans are truly coreferential, which incurs the problem of global inconsistence of coreference cluster. An example which demonstrates the problem of global inconsistence of coreference cluster is as follows.

Chaoyang Road *is a very important artery in the east-west direction. When people living in the west want to cross over from* ***the city****, they have to go via this road. Hence, if a traffic accident occurs at* ***this place****, we can indeed imagine how widespread the extent of the impact will be.*

In above paragraph, mention pairs {*Chaoyang Road*, *this place*} and {*this place*, *the city*} are both locally coreferential, but the cluster of {*Chaoyang Road*, *this place*, *the city*} is not globally coreferential.

4 Proposed Methods

As we mentioned above, the baseline model always makes globally inconsistent decisions, and sometimes gets an incompletely correct cluster because of the independence between these decisions.

Therefore, we propose a coreference cluster modification algorithm to modify the clusters built by the baseline model in this paper. This method confirms the coreference relation between intra-cluster spans which can help rule out the dissimilar span after we get a coreference cluster. Through the modification procedure we can increase the possibility that these spans in a coreference cluster truly refer to the same entity.

We show how to conduct the modification procedure in the situation of n spans in a coreference cluster to demonstrate our algorithm. We first define some variables as follows.

- There are n spans in a coreference cluster and we name them as $\{1, 2, ..., n\}$ in order in a text.
- Span k has a direct link with span i which means span i, as one candidate antecedent of span k, has the highest score among all candidate antecedents.
- We set spans before span k except span i, i.e. $\mathcal{P}(i, k) = \{1, 2, ..., i - 1, i + 1,..., k - 1\}$ as span k 's indirect antecedents. Span k also has indirect links with each span in $\mathcal{P}(i, k)$ which mean extent of compatibility with span k.
- When an incompatible link $s_a(j, k)$ appears, spans before span k except span j form a set $\mathcal{Q}(j, k) = \{1, 2, ..., j - 1, j + 1, ..., k - 1\}$ which is used to consider which span to drop afterwards.

For span k, it originally takes only a direct link $s_a(i, k)$ into consideration while neglects the indirect links with spans in the set $\mathcal{P}(i, k)$. In our method the indirect links within the coreference clusters are labelled explicitly so that we can finally get an enriched coreference cluster full of links between every two spans in the cluster no matter direct or indirect (see Fig. 1).

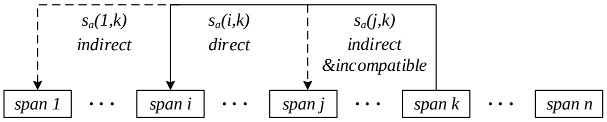

Fig. 1. Enriched coreference cluster after labeling indirect coreference links explicitly.

Value $s_a(i, j)$ can be positive and negative. The greater the abstract value of $s_a(i, j)$ is, the stronger the coreference relation of compatibility (positive) or incompatibility (negative) will be. Furthermore, we design some rules as demonstrated in Algorithm 1. The algorithm, from a high view, can be interpreted from the point of confidence degree because the abstract value of $s_a(i, j)$ represents the extent of coreference relation.

It takes two steps to conduct the algorithm :

- **First step : check.** We need to check whether there is the problem of global inconsistence of coreference cluster. However it is unsafe to judge the relation with the method of directly taking the indirect link into account because the

Algorithm 1 Coreference cluster modification

```
for $k = 3, 4, ..., n$ do
  if $s_a(i,k) + \frac{1}{k-2} \sum_{p \in \mathcal{P}(i,k)} s_a(p,k) < margin$ then
    $j = \arg\min_{p \in \mathcal{P}(i,k)} s_a(p,k)$
    if $\sum_{q \in \mathcal{Q}(j,k)} s_a(q,k) < \sum_{q \in \mathcal{Q}(j,k)} s_a(q,j)$ then
      drop span $k$
    else
      drop span $j$
    end if
  else
    drop none of these spans in a cluster
  end if
end for
```

model has the limitation to represent the coreference relation. To tolerate this kind of mistakes and increase robustness of the model, we introduce the inequity rules that taking direct link and average of all indirect links in $\mathcal{P}(i, k)$ into account to further confirm the coreference relation.

- **Second step : drop.** If the problem of global inconsistence of coreference cluster truly happen, we need to consider which span to drop furthermore. It must be that some indirect link $s_a(j, k)$ is incompatible which means span k and span j are incompatible. We make a comparision between the sum of span j 's links with spans in $\mathcal{Q}(j, k)$ and the sum of span k 's links with spans in $\mathcal{Q}(j, k)$, then we make the modification decision that drop span j or span k.

5 Experiments

5.1 Experimental Setup

We used the English coreference resolution data from the CoNLL-2012 shared task [19] in our experiments. This dataset contains 2802 training documents, 343 development documents, and 348 test documents.

Our model reused the hyperparameters from Lee et al. (2017) [5] so that we can make comparisons with the baseline model. Some parameters of the baseline model are mentioned below.

- **Word representations.** The word embeddings were fixed concatenations of 300-dimensional GloVe embeddings [20] and 50-dimensional embeddings from Turian et al. (2010) [21]. In the character CNN, characters were represented as learned 8-dimensional embeddings. The convolutions had window sizes of 3, 4, and 5 characters, each consisting of 50 filters.
- **Hidden dimensions.** The hidden states in the LSTMs had 200 dimensions.

- **Feature encoding.** All features including speaker, genre, span distance and mention width were represented as learned 20-dimensional embeddings.
- **Pruning.** The baseline model pruned the spans such that the maximum span width L = 10, the number of spans per word $\lambda = 0.4$, and the maximum number of antecedents K = 250.
- **Learning.** The baseline model used ADAM [22] for learning with a minibatch size of 1. 0.5 dropout [23] was applied to the word embeddings and character CNN outputs and 0.2 dropout was applied to all hidden layers and feature embeddings.

5.2 Coreference Cluster Modification

The only hyperparameter in this method is *margin* in the inequilties, which is used to measure the possibility of global inconsistance of coreference cluster. Moreover, some other factors may also affect the performance of our proposed algorithm. We tuned these factors across experiments about different combinations of them on the development dataset as showed in Table 1.

Table 1. Some factors were tuned by experiments on the development dataset, where **number** means the number of spans in a coreference cluster and **function** means the function involved in the first check step.

number	*function*	*margin*	Avg. F1
< 5	mean	0	67.4
< 5	min	0	67.3
< 5	mean	-2	67.6
< 7	mean	-2	**67.7**
< 10	mean	-2	67.6
all	mean	-2	67.3

From this table, we can see that our proposed method still didn't work well for post-processing the clusters with more than 10 spans. The distribution of the size of coreference clusters on the development set given by the baseline model is showed in Fig. 2. We can see that the coreference clusters with less than 10 spans accounted for about 93% of all coreference clusters. Besides, the *mean* function worked slightly better than the *min* function during the check step. One possible reason was that the *mean* function took the information of all indirect links within the cluster into account. Finally, the last row was chosen as the configuration of our proposed method.

5.3 Parameter Tuning

The baseline model simply projects the outputs from the bidirectional LSTMs [18] to a scalar score as we decribe in Equity [5] When computing the weight of

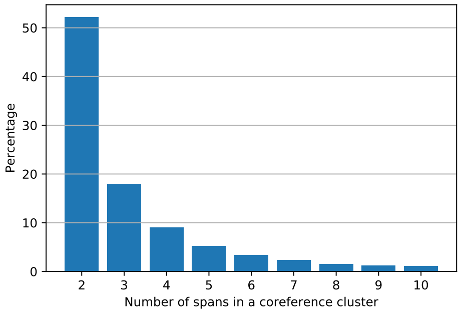

Fig. 2. Distribution of coreference clusters according to number of spans in a coreference cluster.

each word. We replace the simple function with a feed-forward neural network which can help incorporate more information about words in a span to get more accurate attention weights to pick out the head word. The feed-forward neural network in this method consists of two hidden layers with 150 dimensions and rectified linear units [24].

We analysed the error examples of the baseline model on the test dataset. We found that 3934 mentions were not detected, in which 576 mentions had more than 10 words in a span that exceeded the maximum span width.This implied that the model was susceptible to the maximum span width and an accuracy improvement may be achieved by increasing the maximum span width. Therefore, we increased the maximum span width from 10 to 30 words by experiments and obtained the gain of average F1 as shown in Fig. 3.

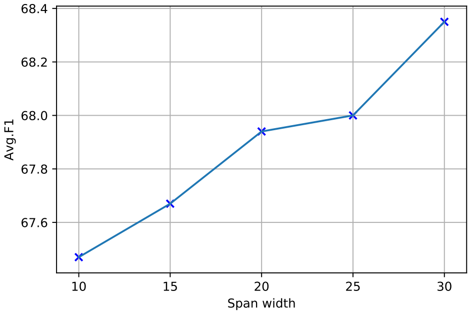

Fig. 3. Average F1 on the test dataset with different maximum width of spans.

5.4 Results

We report the precision, recall, and F1 of the the MUC, B^3, and $CEAF_{\phi4}$ metrics using the official CoNLL-2012 evaluation scripts. The final measurement is the average F1 of the three metrics.

Results on the test set are shown in Table 2. The performances of the systems proposed in the last three years were included for comparison. The baseline model of our methods was the span-ranking model from Lee et al. (2017) [5] which achieved an F1 score of 67.2. Our method achieved an F1 score of 67.5, improving the performance for coreference resolution. Furthermore, we can achieve a higher F1 score of 68.4 after parameter tuning.

Table 2. Results on the test set on the English CoNLL-2012 shared task. The final column (Avg. F1) is the main evaluation metric, computed by averaging the F1 of MUC, B^3, and $CEAF_{\phi4}$.

	MUC			B^3			$CEAF_{\phi4}$			
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Avg. F1
Martschat and Strube (2015) [16]	76.7	68.1	72.2	66.1	54.2	59.6	59.5	52.3	55.7	62.5
Clark and Manning (2015) [13]	76.1	69.4	72.6	65.6	56.0	60.4	59.4	53.0	56.0	63.0
Wiseman et al. (2015) [17]	76.2	69.3	72.6	66.2	55.8	60.5	59.4	54.9	57.1	63.4
Wiseman et al. (2016) [2]	77.5	69.8	73.4	66.8	57.0	61.5	62.1	53.9	57.7	64.2
Clark and Manning (2016b) [4]	79.9	69.3	74.2	71.0	56.5	63.0	63.8	54.3	58.7	65.3
Clark and Manning (2016a) [3]	79.2	70.4	74.6	69.9	58.0	63.4	63.5	55.5	59.2	65.7
Lee et al. (2017) [5]	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
Lee et al. (2018) [25]	81.4	79.5	80.4	72.2	69.5	70.8	68.2	67.1	67.6	73.0
Our proposed	78.3	73.8	76.0	68.3	62.4	65.2	62.8	59.7	61.2	67.5
Our proposed + paramter tuning	79.3	73.9	76.5	70.2	62.7	66.2	63.5	61.2	62.3	68.4

Recently, Lee et al. (2018) [25] has just improved the baseline model by proposing a high-order inference model and tuning some model hyperparameters. The results of this work were also listed in Table 2 for comparison. Although our results were not as good as the ones of Lee et al. (2018) [25], our method has the advantage of simplicity and it can be considered as a rule-based post-processing of the output given by the baseline model.

6 Conclusion

We presented an improved neural coreference resolution method through a cluster modification algorithm which can help modify the coreference cluster to reduce errors caused by global inconsistence of coreference clusters. Additionally, we replace the scoring function with a feed-forward neural network when computing the head word score which can help pick out the most important word. The maximum mention width is also tuned because our experiments showed that the model is susceptible to the maximum width of mentions. Our experimental results demonstrated that these above procedures helped to increase the accuracy of coreference resolution. To improve the performance of the proposed cluster modification algorithm for clusters with large sizes will be a task of our future work.

Acknowledgements. This work was funded in part by Chinese Academy of Sciences President's International Fellowship Initiative (Grant No. 2018VTA0008).

References

1. H. Lee, A. Chang, Y. Peirsman, N. Chambers, M. Surdeanu, and D. Jurafsky, "Deterministic coreference resolution based on entity-centric, precision-ranked rules," *Computational Linguistics*, vol. 39, no. 4, pp. 885–916, 2013.
2. S. Wiseman, A. M. Rush, and S. M. Shieber, "Learning global features for coreference resolution," *arXiv preprint arXiv:1604.03035*, 2016.
3. K. Clark and C. D. Manning, "Deep reinforcement learning for mention-ranking coreference models," *arXiv preprint arXiv:1609.08667*, 2016.
4. ——, "Improving coreference resolution by learning entity-level distributed representations," *arXiv preprint arXiv:1606.01323*, 2016.
5. K. Lee, L. He, M. Lewis, and L. Zettlemoyer, "End-to-end neural coreference resolution," *arXiv preprint arXiv:1707.07045*, 2017.
6. D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.
7. D. Connolly, J. D. Burger, and D. S. Day, "A machine learning approach to anaphoric reference," in *New methods in language processing*, 1997, pp. 133–144.
8. K. Raghunathan, H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky, and C. Manning, "A multi-pass sieve for coreference resolution," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 492–501.
9. G. Durrett and D. Klein, "Easy victories and uphill battles in coreference resolution," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1971–1982.
10. V. Ng and C. Cardie, "Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 2002, pp. 1–7.
11. E. Bengtson and D. Roth, "Understanding the value of features for coreference resolution," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008, pp. 294–303.
12. A. Haghighi and D. Klein, "Coreference resolution in a modular, entity-centered model," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 385–393.
13. K. Clark and C. D. Manning, "Entity-centric coreference resolution with model stacking," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2015, pp. 1405–1415.

14. E. R. Fernandes, C. N. Dos Santos, and R. L. Milidiú, “Latent structure perceptron with feature induction for unrestricted coreference resolution,” in *Joint Conference on EMNLP and CoNLL-Shared Task*. Association for Computational Linguistics, 2012, pp. 41–48.
15. A. Björkelund and J. Kuhn, “Learning structured perceptrons for coreference resolution with latent antecedents and non-local features,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2014, pp. 47–57.
16. S. Martschat and M. Strube, “Latent structures for coreference resolution,” *Transactions of the Association of Computational Linguistics*, vol. 3, no. 1, pp. 405–418, 2015.
17. S. J. Wiseman, A. M. Rush, S. M. Shieber, and J. Weston, “Learning anaphoricity and antecedent ranking features for coreference resolution.” Association for Computational Linguistics, 2015.
18. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
19. S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang, “Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes,” in *Joint Conference on EMNLP and CoNLL-Shared Task*. Association for Computational Linguistics, 2012, pp. 1–40.
20. J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
21. J. Turian, L. Ratinov, and Y. Bengio, “Word representations: a simple and general method for semi-supervised learning,” in *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, 2010, pp. 384–394.
22. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
23. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
24. V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.
25. K. Lee, L. He, and L. Zettlemoyer, “Higher-order coreference resolution with coarse-to-fine inference,” *arXiv preprint arXiv:1804.05392*, 2018.