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

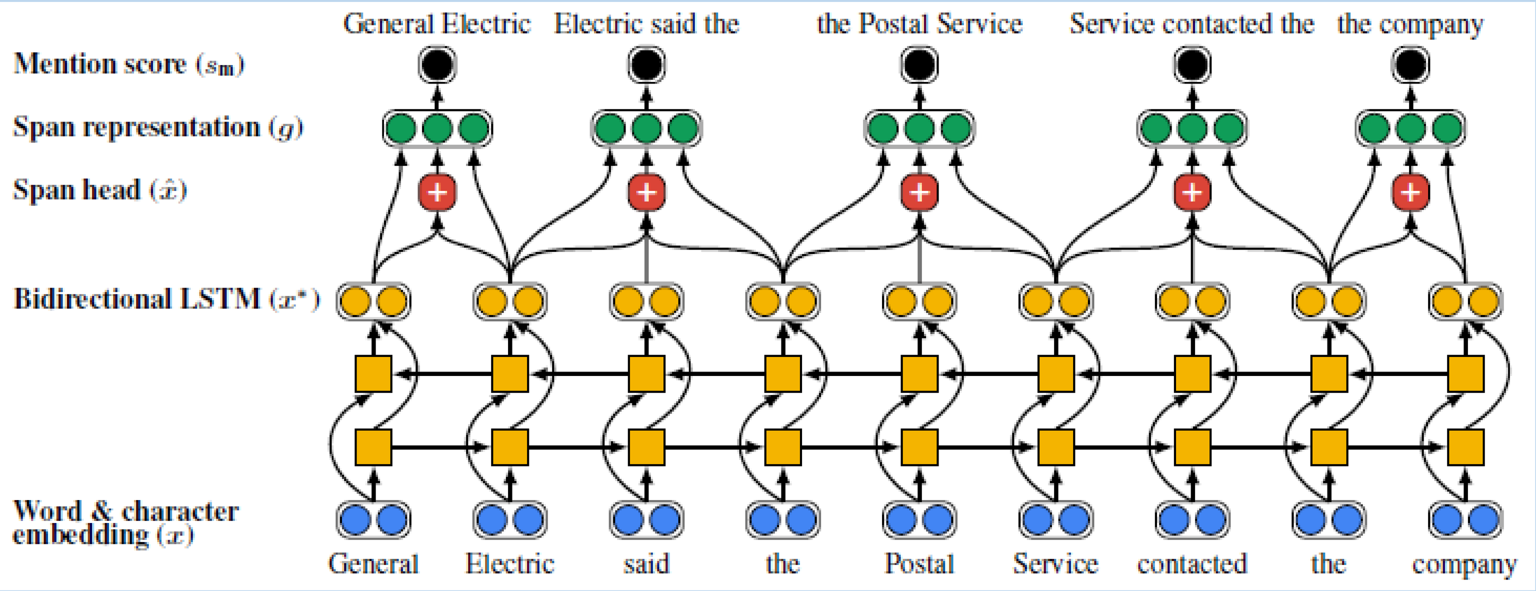
Jia-Chen Gu, Zhen-Hua Ling, and Nitin Indurkhya
National Engineering Laboratory for Speech and Language Information Processing,
University of Science and Technology of China
gujc@mail.ustc.edu.cn, {zhling, nitin}@ustc.edu.cn

Introduction

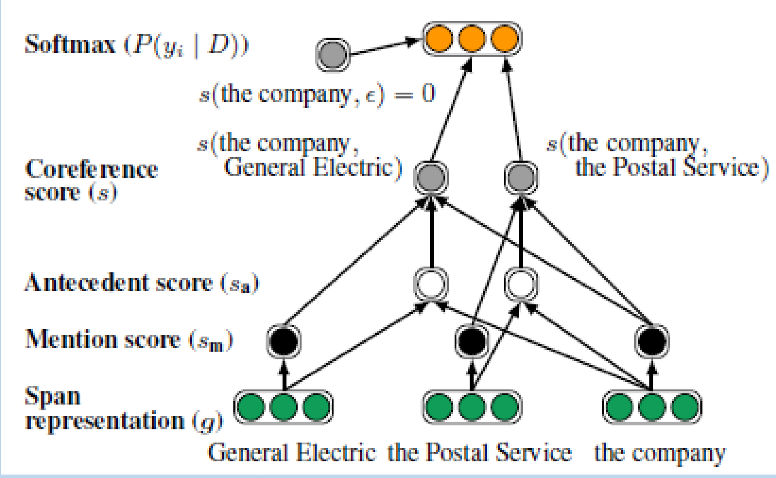
- **Coreference resolution**
  - **Definition**: the task of finding all expressions that refer to the same real-world entity in a text or dialogue
  - **Example**: "I voted for Nader because he was most aligned with my values," she said.
- **Methods for coreference resolution**
  - Mention-pair classifiers (Bengtson et al., 2008)
  - Entity-level models (Clark and Manning, 2016)
  - Latent-tree models (Martschat and Strube, 2015)
  - Mention-ranking models (Wiseman et al., 2015)
  - Span-ranking models (Lee et al., 2017)
    - Formulate the task as a set of antecedent assignments for each span
    - First end-to-end neural model for coreference resolution
    - Not rely on syntactic parsers and many hand-engineered features
    - Make independent decisions about whether two mentions are coreferential and then establish a coreference cluster through this kind of coreference relation

Methods

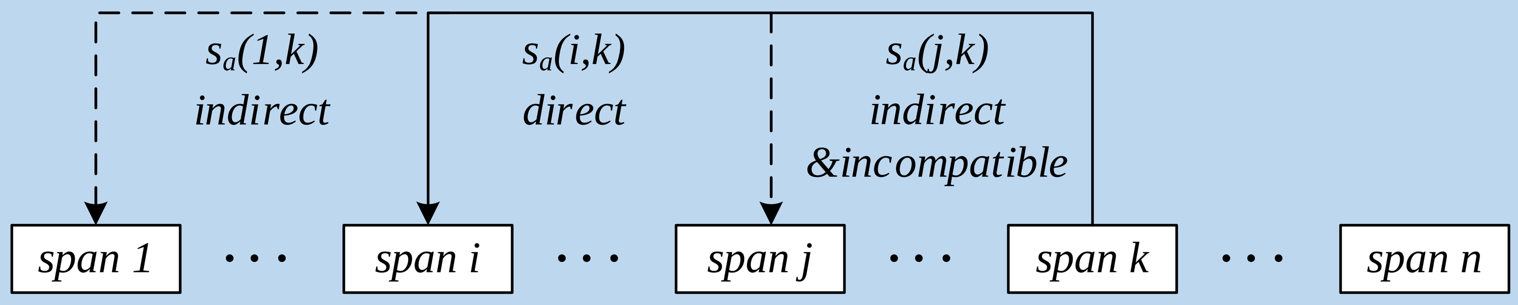
- Model overview (Lee et al., 2017)
  - Compute embedding representations of spans for scoring potential mentions



  - Compute antecedent scores from pairs of span representations



- To alleviate the problem of **global inconsistence**, we propose a **coreference cluster modification algorithm** to confirm the coreference relation between intra-cluster spans which can help rule out the dissimilar span after we get a coreference cluster.
  - **First step : check**. Check whether there is the problem of global inconsistence of coreference cluster.
  - **Second step : drop**. If the problem of global inconsistence of coreference cluster truly happen, we need to consider which span to drop furthermore.



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

- We tune the hyperparameters from two aspects
  - Experiments show the model is susceptible to the maximum span width.
  - Computing the weight of each word to form a weighted sum of word vectors in a span with a feed-forward neural network, which can help get more accurate attention weights to pick out the head word.

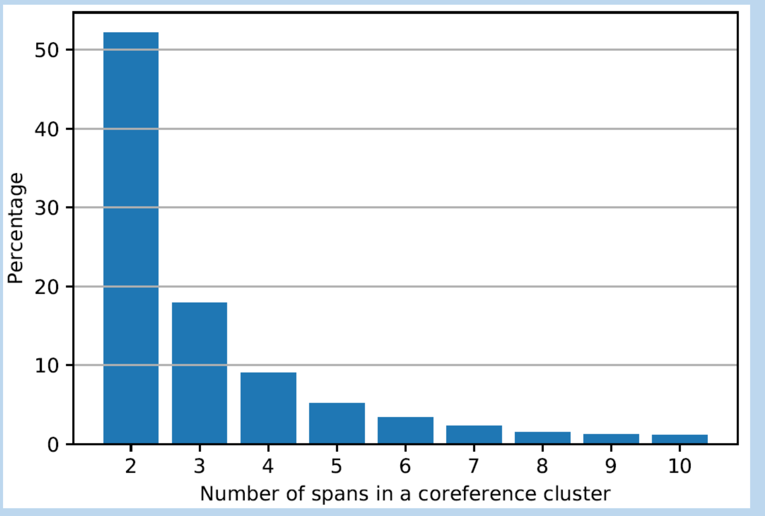
Dataset

- **CoNLL-2012 shared task**
  - English coreference resolution corpus
  - Contains 2802 training documents, 343 development documents, and 348 test documents.

Experiments

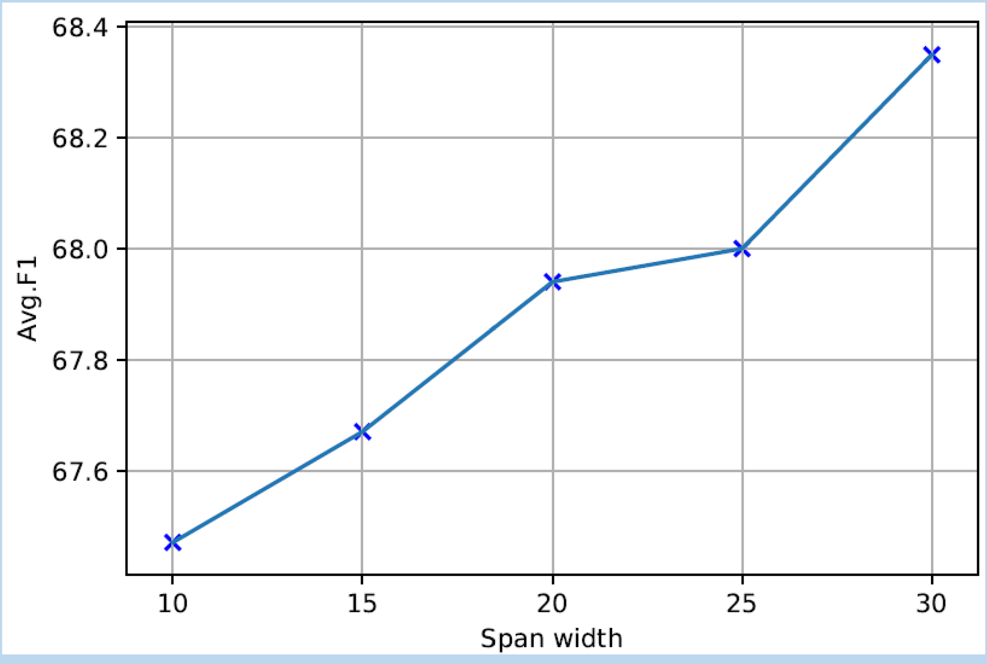
Margin tuning on development dataset

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



- The only hyperparameter in our method is *margin* in the inequilties, which is used to measure the possibility of global inconsistance of coreference cluster.
- The coreference clusters with less than 10 spans accounted for about 93% of all coreference clusters.

Avg.F1 on test dataset with different maximum spans width



- 3934 mentions were not detected, in which 576 mentions had more than 10 words in a span that exceeded the maximum span width, taking a large part in the errors because of the limitation of the maximum span width.

Results on the test set on the English CoNLL-2012 shared task

	MUC			B^3			$CEAF_{\phi 4}$			
	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

- The baseline model of our methods was the span-ranking model from Lee et al. (2017) 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.
- 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.

Conclusion

- We proposed a cluster modification algorithm which can help modify coreference clusters to reduce errors caused by global inconsistence of coreference clusters.
- Our experiments show that the model is susceptible to the maximum mention width which can help to increase the accuracy of coreference resolution.
- We replace the scoring function with a feed-forward neural network which can help pick out the most important word.

References

1. Bengtson, Eric, and Dan Roth. "Understanding the Value of Features for Coreference Resolution." Urbana 51: 61801.
2. Clark, Kevin, and Christopher D. Manning. "Improving Coreference Resolution by Learning Entity-Level Distributed Representations."
3. Martschat, Sebastian, and Michael Strube. "Latent structures for coreference resolution." Transactions of the Association of Computational Linguistics 3.1 (2015): 405-418.
4. Wiseman, Sam, et al. "Learning Anaphoricity and Antecedent Ranking Features for Coreference Resolution."
5. Lee, Kenton, et al. "End-to-end Neural Coreference Resolution." Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017.