LINGMESS: Linguistically Informed Multi Expert Scorers for Coreference Resolution

Shon Otmazgin[1] **Arie Cattan**[1] **Yoav Goldberg**[1,2]
[1]Computer Science Department, Bar Ilan University
[2]Allen Institute for Artificial Intelligence
{shon711,arie.cattan,yoav.goldberg}@gmail.com

Abstract

While coreference resolution typically involves various linguistic challenges, recent models are based on a single pairwise scorer for all types of pairs. We present LINGMESS, a new coreference model that defines different categories of coreference cases and optimize multiple pairwise scorers, where each scorer learns a specific set of linguistic challenges. Our model substantially improves pairwise scores for most categories and outperforms cluster-level performance on Ontonotes.[1]

1 Introduction

Coreference resolution is the task of clustering textual mentions that refer to the same discourse entity. This fundamental task requires many decisions. In this work, we argue that different *kinds* of decisions involve different challenges. To illustrate that, consider the following text:

*"**Lionel Messi** has won a record seven Ballon d'Or awards. **He** signed for Paris Saint-Germain in August 2021. "**I** would like to thank **my** family", said **the Argentinian footballer**. **Messi** holds the records for most goals in La Liga"*

To correctly identify that the pronoun "He" refers to "Lionel Messi", models need to model the discourse, while linking "my" to "I" may rely more heavily on morphological agreement. Likewise, linking "the Argentinian footballer" to "Lionel Messi" requires world knowledge, while linking "Messi" to "Lionel Messi" may be achieved by simple lexical heuristics.

Despite these inherent differences, recent models are based on a *single* pairwise scorer for all types of pairs, regardless of the different challenges that need to be addressed (Lee et al., 2017, 2018; Joshi et al., 2019; Kantor and Globerson, 2019; Joshi et al., 2020; Xu and Choi, 2020; Xia et al., 2020; Toshniwal et al., 2020; Thirukovalluru et al., 2021; Kirstain et al., 2021; Cattan et al., 2021; Dobrovolskii, 2021).

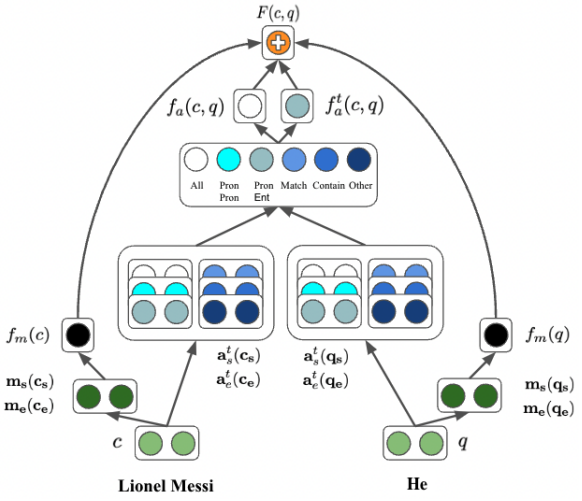

Figure 1: Architecture of our multi-head expert model. Given two spans *"Lionel Messi"* and *"He"*, the contextualized vectors (green) are fed into $\mathbf{m_s}(\cdot)$ and $\mathbf{m_e}(\cdot)$ to compute mention scores $f_m(\cdot)$ (black), and to $\mathbf{a_s^t}(\cdot)$, $\mathbf{a_e^t}(\cdot)$ to perform per category antecedent scores (blue family). The relevant category (Pron-Ent) score $f_a^t(\cdot,\cdot)$ (turquoise) and the general score $f_a(\cdot,\cdot)$ (white). The final score $F(c,q)$ is the sum of these four scores.

In this work, we identify a set of linguistically meaningful classes of decisions: (a) linking pronouns to pronouns (PRON-PRON); (b) linking pronouns to entities (ENT-PRON); (c) linking entities which share the exact lexical form (MATCH); (d) linking entities where the lexical form of one contains the lexical form of the other (CONTAINS); (e) other cases. Each of these classes is easy to identify deterministically, each contains both positive and negative instances, and each could benefit from a somewhat different decision process. An example of each class is given in Table 1.

We present **Ling**uistically Informed **M**ulti **E**xpert **S**corers (LINGMESS), a coreference model

[1]Our model is available in https://github.com/shon-otmazgin/lingmess-coref

Phenomenon	Positive	Negative
PRON-PRON	*A couple of **my** law clerks were going to ... and **I** was afraid **I** was going to...*	***I** have seen one or two men die, bless **them**.*
ENT-PRON	***Spain, Argentina, Thailand and Indonesia** were doing too little to prevent ... across **their** borders.*	*Tonight, to kick off **the effort**, CNN will premiere **its** first prime - time newscast in years.*
MATCH	*... says Paul Amos, **CNN** executive vice president for programming. Accordingly, **CNN** is ...*	*Hertz and Avis can not benefit **Budget's** programs, " said Bob Wilson, **Budget's** vice president ...*
CONTAINS	*He reportedly showed **DeLay** a videotape that made him weep. **Tom DeLay** then ...*	*Give **SEC** authority to halt securities trading, (also opposed by new **SEC chairman**) ...*
OTHER	*They also saw **the two men who were standing with him**. When **Moses and Elijah** were leaving ...*	***The company** is already working on its own programming ... **the newspaper** said.*

Table 1: Example of each category, taken from Ontonotes development set. The mentions pairs are in bold. We define the categories as follows. PRON-PRON: the two spans are pronouns, ENT-PRON: a pronoun and another span, MATCH: two non-pronoun spans that have the same string, CONTAINS: two spans such that one contains the other, OTHER: all other pairs.

which categorizes each pairwise decision into one of these classes,[2] and learns a separate scoring function for each class. Specifically, we extend the recent *s2e*'s model (Kirstain et al., 2021) by adding per-category scoring, but the method is general and may work with other coreference models as well. As illustrated in Figure 1, the final coreference score between two spans is composed—in addition to the individual mention scores—of two scores: a general antecedent-compatibility score and an "expert" antecedent compatibility score which depends on the linguistic-type of the pair.

We show that this significantly improves the pairwise F1 scores, and also reflected in a 1-point increase in cluster-level CoNLL F1 score on Ontonotes (Pradhan et al., 2012).

We also inspect the performance of the model for each category separately, showing that some classes improve more than others. This analysis further provides a finer-grained understanding of the models and the coreference phenomena, and points out directions for future research.

2 Background: the s2e Model

The *s2e* model (Kirstain et al., 2021) is a lightweight and efficient coreference model, which avoids the costly construction of full-fledged span representations, by considering only span boundaries, and also achieves the current best coreference scores among all practical models.[3]

Given a sequence of tokens $x_1, ..., x_n$, each span is represented by its start and end tokens. For a pair of spans $c = (x_i, x_j)$, $q = (x_k, x_\ell)$ where c ("candidate") appears before q ("query"), a parameterized function $f_m(c)$ ($f_m(q)$) scores how fit a given span is to be a mention, while a parameterized function $f_a(c, q)$ scores how fit c is to be a coreferring antecedent of q (As computing antecedent scores for all possible pairs would result in a complexity of $\mathcal{O}(n^4)$, in practice only the highest scoring spans according to f_m are fed to the antecedent scorer f_a). These functions operate over contextualized vector representations, obtained by a BERT-like model. For the exact function form, see (Kirstain et al., 2021).

Finally, similarly to Lee et al. (2017), the final pairwise score $F_g(c, q)$ is composed of the two mention scores $f_m(q)$, $f_m(c)$ and the antecedent score $f_a(c, q)$.

$$F_g(c,q) = \begin{cases} f_m(c) + f_m(q) + f_a(c,q) & c \neq \varepsilon \\ 0 & c = \varepsilon \end{cases}$$

where ε is the null antecedent.

For each possible mention q, the learning objective optimizes the sum of probabilities over the true antecedent $\hat{c}$ of q:

$$L_g(q) = \log \sum_{\hat{c} \in \mathcal{C}(q) \cap \text{GOLD}(q)} P_g(\hat{c} \mid q)$$

where $\mathcal{C}(q)$ is the set of all candidate antecedents[4] together with a null antecedent ε. $\text{GOLD}(q)$ is the

[2] Others—more fine-grained—distinctions are of course also possible, but we leave exploration of them to future work.

[3] Dobrovolskii (2021)'s model slightly surpasses the *s2e* model but requires more GPU memory for training. The CorefQA model (Wu et al., 2020) achieves a substantially higher score, but also requires to run a separate BERT inference for each mention, making it highly impractical.

[4] All spans before q that passed some pruning threshold.

set of the true antecedents of q. $P_g(c \mid q)$ is computed as a softmax over $F_g(c, q)$ scores for c values in $\mathcal{C}(q)$:

$$P_g(c \mid q) = \frac{\exp F_g(c, q)}{\sum_{c' \in \mathcal{C}(q)} \exp F_g(c', q)}$$

3 LINGMESS

Clustering coreferring entities typically involves many different phenomena, which we argue should be addressed in a different manner. Indeed, linking two string match entities such as *Hong Kong* to *Hong Kong* is different from linking an entity to a pronoun, for instance, *Lionel Messi* and *he*. In the string match case, the mentions tokens are alone indicative features for the linking decision, (though there are cases where the lexical identify should be ignored, such as *Washington* the U.S government vs. *Washington* the city), whereas the entity-pronoun case requires a different kind of discourse analysis. Therefore, our core contribution is proposing to allocate a dedicated scorer $f_a^t(c, q)$ for each phenomenon type t, in addition to the general antecedent scorer $f_a(c, q)$. The overall architecture of our model is shown in Figure 1.

Concretely, we extend the *s2e* model with five additional antecedent scorers $f_a^t(\cdot, \cdot)$ where $t \in \{\text{PRON-PRON}, \text{ENT-PRON}, \text{MATCH}, \text{CONTAINS}, \text{OTHER}\}$, the five categories we list in Table 1.

The pairwise scoring function now becomes:

$$F(c, q) = \begin{cases} f_m(c) + f_m(q) + f(c, q) & c \neq \varepsilon \\ 0 & c = \varepsilon \end{cases} \quad (1)$$

$$f(c, q) = f_a(c, q) + f_a^{T(c,q)}(c, q)$$

where $T(c, q)$ is a rule-based function to determine the category t of the pair (q, c). $F(c, q)$ is the final score function, a sum of four scores, $f_m(q)$ how likely is span q being a mention, $f_m(c)$ how likely is span c being a mention, $f_a(c, q)$ is the "general" scorer, how likely is c is an antecedent of q, and lastly, $f_a^t(c, q)$ is the "expert" scorer for the category t, how likely is c is an antecedent of q. Each of the six pairwise scoring functions (f_a and the fine f_a^t) is parameterized separately using its own set of matrices. The transformer-based encoder and the mention scorer are shared between all the different pairwise scorers.

Learning Through training, for each span q, our model optimizes the objective function L_{coref} over the sum of probabilities of all true antecedents of q:

$$L_{coref}(q) = \log \sum_{\hat{c} \in \mathcal{C}(q) \cap \text{GOLD}(q)} P(\hat{c} \mid q)$$

Here, $P(\hat{c} \mid q)$ is a softmax over $F(\hat{c}, q)$ scores, that is our new score function described in Figure 1.

$$P(\hat{c} \mid q) = \frac{\exp F(\hat{c}, q)}{\sum_{c' \in \mathcal{C}(q)} \exp F(c', q)}$$

This model is also the one used in inference. However, this objective does not explicitly push each category ("expert") to specialize. For example, for the PRON-PRON cases, it would be useful to explicitly train the model to distinguish between the possible antecedents of that type (without regarding other antecedents), as well as to explicitly distinguish between a pronoun antecedent and a null antecedent. To this end, we extend the training objective by also training each expert separately:

$$L_t(q) = \log \sum_{\hat{c} \in \mathcal{C}_t(q) \cap \text{GOLD}(q)} P_t(\hat{c} \mid q)$$

$$F_t(c, q) = \begin{cases} f_m(c) + f_m(q) + f_a^t(c, q) & c \neq \varepsilon \\ 0 & c = \varepsilon \end{cases}$$

$$P_t(\hat{c} \mid q) = \frac{\exp F_t(\hat{c}, q)}{\sum_{c' \in \mathcal{C}_t(q)} \exp F_t(c', q)}$$

Note that for $L_t(q)$ we replace $\mathcal{C}(q)$ with $\mathcal{C}_t(q)$, considering only the potential antecedents that are compatible with the span q for the given type (for example, for $L_{\text{PRON-PRON}}$ and a span q corresponding to a pronoun, we will only consider candidates c which appear before q and are also pronouns).

Our final objective for each mention span q is thus:

$$L(q) = L_{coref}(q) + L_{tasks}(q)$$
$$L_{tasks}(q) = \sum_t L_t(q) + L_g(q)$$

Inference We form the coreference chains by linking each mention q to its most likely antecedent c according to $F(c, q)$ (Eq. 1). We do not use higher-order inference as it has been shown to have a marginal impact (Xu and Choi, 2020).

	MUC			B^3			$CEAF_{\phi_4}$			
	P	R	F1	P	R	F1	P	R	F1	Avg. F1
Joshi et al. (2020)	85.8	84.8	85.3	78.3	77.9	78.1	76.4	74.2	75.3	79.6
Kirstain et al. (2021)	86.5	85.1	85.8	80.3	77.9	79.1	76.8	75.4	76.1	80.3
LINGMESS	**87.6**	**85.3**	**86.5**	**82.1**	**78.6**	**80.3**	**78.0**	**76.0**	**77.0**	**81.3**

Table 2: Performance on the test set of the English OntoNotes 5.0 dataset. The averaged F1 of MUC, B^3, $CEAF_\phi$ is the main evaluation metric.

	Kirstain et al. (2021)			LINGMESS		
	P	R	F1	P	R	F1
PRON-PRON	91.0	74.1	81.7	90.0	87.8	**88.9**
ENT-PRON	82.0	74.1	**77.9**	80.1	74.7	77.3
MATCH	92.6	88.1	90.3	92.8	93.2	**93.0**
CONTAIN	70.8	79.5	74.9	75.0	82.1	**78.4**
OTHER	62.1	72.1	66.7	70.5	69.3	**69.9**

Table 3: Pairwise performance by category, on the dev set of the English OntoNotes 5.0 dataset. For each pair c and q a positive prediction is a score greater than 0, positive label is if c is an true antecedent of q.

4 Experiments

In our experiments, we use the English OntoNotes 5.0 dataset (Pradhan et al., 2012) to train and evaluate our model. This dataset contains 2802 documents for training, 343 for development, and 348 for test. To implement our categories method, we extend the *s2e*'s implementation based on PyTorch (Paszke et al., 2019) and Hugging face Transformers library (Wolf et al., 2020).

Baseline We consider Kirstain et al. (2021) *s2e* model as our baseline, which is an efficient baseline that achieves 80.3 F1 on OntoNotes test set.

Hyperparameters We used the same hyperparameters as the baseline except the size of the feed forward neural network (FFNN) used by the functions $f_m(\cdot)$ and $f_a(\cdot,\cdot)$. The FFNN size of the baseline is 3072 parameters. Our method introduce $f_a^t(\cdot,\cdot)$ function for each category t, thus, to fit into memory we reduce the size of FFNN to 2048 parameters. Similar to the baseline our head method is on top of Longformer-Large (Beltagy et al., 2020), resulting in a total of 569M learnable parameters, comparable size to the baseline which contains 494M learnable parameters.

Performance Table 2 presents the performance of LINGMESS in comparison to the baseline with the standard evaluation metrics for coreference resolution: MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and $CEAF_{\phi_4}$ (Luo, 2005). The main evaluation is the average F1 of the three metrics. LINGMESS outperforms previous baselines according to all evaluation metrics. The CoNLL F1 on the development set is 81.4.

Advantages of Categories To assess that the improvement of LINGMESS is due to the decomposition into our set of categories and not due to the added parameters, we did two experiments. First, we train a random baseline, which randomly assigns a category for each pair.[5] Second, we train our model by optimizing only the overall loss L_{coref} and not L_{tasks}. In both experiments, we obtain similar results to the baseline, likely due to the dominance of the non-category parameters.

In addition to the standard coreference evaluation, we measure pairwise performance and report the results for each category. Given a mention-pair (q, c), if $F(c, q)$ is greater than 0, we treat it as a positive score, otherwise negative. We then measure precision, recall and F1 based on gold clusters labels. Table 3 shows the pairwise performance of the *s2e* model and LINGMESS. LINGMESS outperforms *s2e* by a significant margin for all categories (e.g +7.2 F1 for PRON-PRON, +3.9 F1 for CONTAIN, etc.) except for ENT-PRON where the *s2e* model surpasses LINGMESS by only 0.6 F1. The gain in coreference metrics is not so significant because the coreference chains are formed by linking a mention to only one antecedent with the highest $F(c, q)$. Nonetheless, LINGMESS provides scores with higher quality which can be useful for injecting coreference signals in downstream tasks.

We note also that models address each category differently. Indeed, LINGMESS achieves 88.9 F1 on PRON-PRON and 93.0 F1 on MATCH, whereas it achieves only 78.4 for CONTAIN and 69.9 F1 for OTHER. These results indicate the vast room for improvement in the categories with lowest scores.

[5] For each pair of mentions (c, q), we take the modulo of the sum of the ASCII code of the last token of c and q.

5 Related Work

The deterministic sieve-based model (Lee et al., 2013) is an early system that breaks down the coreference decisions into multiple linguistic categories. They adopt an easy-first approach where coreference decisions in the first sieves help disambiguate later decisions. Lu and Ng (2020) analyze empirically the performance of recent coreference resolvers on various fine-grained resolution classes of mentions (e.g gendered pronoun vs. 1st and 2nd pronoun, etc). Our work makes progress in that direction by optimizing separately a supervised model for different categories of mentions.

6 Conclusion

We propose LINGMESS, an approach for coreference resolution that learns a separate antecedent scorer for different classes of coreference cases. LINGMESS outperforms the baseline on Ontonotes according to both cluster-level and pairwise F1 scores. These results demonstrate that optimizing separately the different linguistic challenges of a general NLP task is an appealing approach for improving performance.

References

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 79–85, Montreal, Quebec, Canada. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv*, abs/2004.05150.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2021. Cross-document coreference resolution over predicted mentions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5100–5107, Online. Association for Computational Linguistics.

Vladimir Dobrovolskii. 2021. Word-level coreference resolution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7670–7675, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5803–5808, Hong Kong, China. Association for Computational Linguistics.

Ben Kantor and Amir Globerson. 2019. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677, Florence, Italy. Association for Computational Linguistics.

Yuval Kirstain, Ori Ram, and Omer Levy. 2021. Coreference resolution without span representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 14–19, Online. Association for Computational Linguistics.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.

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*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

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, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Jing Lu and Vincent Ng. 2020. Conundrums in entity coreference resolution: Making sense of the state of the art. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6620–6631, Online. Association for Computational Linguistics.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang,

Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

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*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Raghuveer Thirukovalluru, Nicholas Monath, Kumar Shridhar, Manzil Zaheer, Mrinmaya Sachan, and Andrew McCallum. 2021. Scaling within document coreference to long texts. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3921–3931, Online. Association for Computational Linguistics.

Shubham Toshniwal, Sam Wiseman, Allyson Ettinger, Karen Livescu, and Kevin Gimpel. 2020. Learning to Ignore: Long Document Coreference with Bounded Memory Neural Networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8519–8526, Online. Association for Computational Linguistics.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6953–6963, Online. Association for Computational Linguistics.

Patrick Xia, João Sedoc, and Benjamin Van Durme. 2020. Incremental neural coreference resolution in constant memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8617–8624, Online. Association for Computational Linguistics.

Liyan Xu and Jinho D. Choi. 2020. Revealing the myth of higher-order inference in coreference resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8527–8533, Online. Association for Computational Linguistics.