Understanding Mention Detector-Linker Interaction in Neural Coreference Resolution

Zhaofeng Wu
Paul G. Allen School of CSE
University of Washington
zfw7@cs.washington.edu

Matt Gardner
Allen Institute for AI
mattg@allenai.org

Abstract

Despite significant recent progress in coreference resolution, the quality of current state-of-the-art systems still considerably trails behind human-level performance. Using the CoNLL-2012 and PreCo datasets, we dissect the best instantiation of the mainstream end-to-end coreference resolution model that underlies most current best-performing coreference systems, and empirically analyze the behavior of its two components: mention detector and mention linker. While the detector traditionally focuses heavily on recall as a design decision, we demonstrate the importance of precision, calling for their balance. However, we point out the difficulty in building a precise detector due to its inability to make important anaphoricity decisions. We also highlight the enormous room for improving the linker and show that the rest of its errors mainly involve pronoun resolution. We propose promising next steps and hope our findings will help future research in coreference resolution.

1 Introduction

Coreference resolution identifies mentions in a document that co-refer to the same entity. It is an important task facilitating many applications such as reading comprehension (Dasigi et al., 2019) and text summarization (Azzam et al., 1999).

Lee et al. (2017) proposed the first neural end-to-end architecture for coreference resolution. Most recent systems use it as a backbone and employ better scoring functions (Zhang et al., 2018), pruning procedures (Lee et al., 2018), or token representations (Joshi et al., 2019, 2020).[1] Despite this usage, little in-depth analysis has been done to better understand the inner workings of such an influential system. Xu and Choi (2020) analyzed the effect of the high-order inference, while Subramanian and Roth (2019) and Zhao et al. (2018) respectively examined its generalizability and gender bias. Few work has inspected the interaction between its components. Lu and Ng (2020) conducted oracle experiments that are related to ours, but without fine-grained control over confounding factors affecting oracle mentions. Such an understanding is important: for example, Kummerfeld and Klein (2013)'s dissection of the then-best classical coreference systems inspired many important follow-up works (Peng et al., 2015; Martschat and Strube, 2015; Wiseman et al., 2016, *inter alia*). However, it is unknown if observations on such classical feature-based and often pipelined systems extend to current neural end-to-end models.

We consider the best instantiation of this model family, SpanBERT (Joshi et al., 2020) + c2f-coref (Lee et al., 2018), and investigate the interaction between its two components: mention detector and mention linker. We study how their errors independently or jointly affect the final clustering.

Using the CoNLL-2012 (Pradhan et al., 2012) and PreCo (Chen et al., 2018) datasets, we highlight the low-precision, high-recall nature of the detector. While traditionally only recall is emphasized for the detector as a design decision (Lee et al., 2011; Lee et al., 2017), we show huge degradation from noisy mentions and, perhaps surprisingly, increasing the number of candidates considered by the baseline linker only deteriorates the performance. While some classical coreference pipelines focused on detector precision (Uryupina, 2009), it is rarely emphasized for current end-to-end systems. We hence stress the importance of a precision-recall balance for the detector and demonstrate how pruning hyperparameters, in addition to reducing computational complexity, control this trade-off. However, we show the difficulty of obtaining a precise detector by demonstrating the importance of anaphoricity decisions and the inability of the detector to make them. Finally, we highlight the high potential of the linker and that the

[1] Except Wu et al. (2020) which has not seen wide adoption.

Proceedings of the 4th Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2021), pages 150–157
Punta Cana, Dominican Republic, November 10–11, 2021. ©2021 Association for Computational Linguistics

remaining errors mainly involve pronoun resolution. We hope this work sheds light on the internals of the mainstream coreference system and, with our proposed next steps, catalyze future research. We believe some of our findings may also transfer to other tasks with a similar joint span detection and span (pair) classification architecture, such as SRL (He et al., 2018), IE (Luan et al., 2019), and entity linking (Kolitsas et al., 2018). See Jiang et al. (2020) which subsumes many other tasks under such a span-based framework.

2 Background

Model We study the coarse-to-fine coreference system (c2f-coref; Lee et al. 2018). It assigns an antecedent for every span in a document of length T, including a dummy that indicates non-mentions or non-anaphoric mentions. The final clustering is the transitive closure of connected spans. The system consists of a mention detector and a mention linker. The detector scores all $O(T^2)$ spans up to length L and outputs the λT highest-scoring spans as possibly anaphoric mentions. The linker links each mention candidate with the highest-scoring antecedent among K ones. Hyperparameters L, λ, and K control the number of considered spans and antecedents, reducing computational complexity.

Data CoNLL-2012 is the most common dataset to test coreference models. However, it lacks singleton mention annotation (Pradhan et al., 2012).

Singleton, or non-anaphoric, mentions do not co-refer with other spans, e.g. "The dog" in "[The dog] barks." However, they may become anaphoric in another context, e.g. "[The dog] barks at [itself]." Being a mention is a span's inherent property, while anaphoricity, whether or not a mention co-refers, is context-dependent. We use "all mentions" to refer to the union of singleton and anaphoric mentions.

To understand the effect of singleton mentions, we heuristically generate all mentions for CoNLL-12 (§B) for relevant experiments. We also experiment with PreCo, a coreference dataset with annotated singleton mentions. We do all analyses on development sets and report dataset statistics in §A.

3 Experiments

Settings We embed tokens with SpanBERT-large, a pre-trained transformer (Vaswani et al., 2017) with state-of-the-art performance in coreference resolution. We choose $L = 30, \lambda = 0.4, K = 50$. We only keep the first 110 sentences per document during training. To reduce confounding factors, we do not use speaker and genre metadata.

	CoNLL-12	PreCo
Coref F_1	79.17	85.04
ANA. P	28.37	39.23
ANA. R	96.42	98.40
ALL P	82.04	76.55
ALL R	57.35	95.98

Table 1: Original system coreference F_1 and precision / recall for anaphoric mentions (ANA.) and all mentions (ALL) on CoNLL-12 and PreCo development sets.

"Original" System refers to a standard SpanBERT + c2f-coref trained baseline. Its F_1 score[2] is reported in Table 1, similar to the results in Joshi et al. (2020) considering we disregard metadata.

Oracles We build oracle detectors where, starting from the original system's mention candidates (its detector output), we either remove all non-gold mentions (prefect precision), add all missing gold mentions (perfect recall), or both (perfect precision & recall). We give the altered, rather than the original, mention candidates to the linker. We consider both anaphoric mentions and all mentions as gold mentions and modify either in a post-hoc manner or re-train the system with the altered candidates. To control for a non-trainable detector, we train only a linker reusing the original system's mention candidates, dubbed **Fixed Detector**. We consider this baseline as the comparison target for the oracles. Besides oracle detectors, we also build an oracle linker that assigns the correct antecedent (including dummy) to each of the λT mention candidates.

4 Precision-Recall Trade-Off for the Mention Detector

Traditionally, coreference systems heavily favor recall over precision for the detector (Lee et al., 2011) as the linker cannot recover missed mentions. Similarly, our c2f-coref system gets >96% anaphoric mention recall yet only <40% precision (Table 1). We therefore explore if detector recall is always more important than its precision. If more spans are considered by increasing the max span width L or the number of spans considered per word λ, will the system performance necessarily improve? In the extreme case, if we hypothetically

[2]We use coreference F_1 to refer to the average F_1 of MUC, B^3, and $CEAF_{\phi_4}$, the most common coreference metric.

		Span Error	Conflated Entities	Extra Mention	Extra Entity	Divided Entity	Missing Mention	Missing Entity
Original		1.6	3.1	1.3	3.1	2.6	2.1	4.5
Fixed Detector		1.7	3.1	1.7	3.3	2.8	2.1	4.7
ANA.	Perfect P	0.0	2.4	0.0	0.0	2.0	2.6	5.7
Post-hoc	Perfect R	1.6	3.1	1.4	3.1	2.6	2.0	4.4
Oracle	Perfect P&R	0.0	2.3	0.0	0.0	1.9	2.5	5.5
ANA.	Perfect P	0.0	3.4	0.0	0.0	0.9	1.5	1.4
Re-train	Perfect R	0.7	3.1	2.0	3.4	2.6	1.5	4.4
Oracle	Perfect P&R	0.0	3.0	0.0	0.0	1.0	0.4	0.3

Table 2: The F_1 score improvement after fixing different types of errors on the CoNLL-12 development set. The errors are independently fixed after span errors are fixed. The categorization is from Kummerfeld and Klein (2013).

		CoNLL-12	PreCo
Fixed Detector		78.28	84.64
ANA.	Perfect P	86.02	90.31
Post-hoc	Perfect R	79.37	85.17
Oracle	Perfect P&R	86.28	90.45
ANA.	Perfect P	89.98	95.09
Re-train	Perfect R	79.65	85.22
Oracle	Perfect P&R	92.39	96.50
ALL	Perfect P	79.48	88.37
Re-train	Perfect R	78.52	85.23
Oracle	Perfect P&R	80.05	89.13
Oracle Linker		97.07	98.69

Table 3: Baseline and oracle coreference F_1 for anaphoric mentions (ANA.) and all mentions (ALL) on CoNLL-12 and PreCo development sets. "Fixed Detector" is the baseline with a non-trainable detector. The middle three sections are oracle detectors with perfect candidate precision/recall. The last row is an oracle linker that always makes correct antecedent decisions.

had enough compute that allows the linker to consider all $O(T^4)$ span-antecedent pairs, should we simply remove the pruning in the detector?

The Aggregated Importance of Precision For all oracles in Table 3, fixing precision yields a larger improvement than recall, especially with anaphoric mentions. This highlights the importance of detector precision and the extent to which the linker suffers from noisy mention candidates. In Table 2, we present the F_1 improvement after independently fixing categorized errors following Kummerfeld and Klein (2013).[3] Noisy candidates result in extra mention and extra entity errors, fixing which accounts for more than half of the ≈8 F_1 gap between the post-hoc perfect precision oracle and the baseline for CoNLL-12 (Table 3). Furthermore, re-training the system to leverage the distributional shift of the absence of noise leads to another ≈4 and 5 F_1 increase (CoNLL-12/PreCo).

To analyze how higher detector precision helps the linker, we examine the coreference score the linker assigns to every span-antecedent pair. The anaphoric mention re-trained perfect precision oracle has an average score of –13.0 on CoNLL-12, higher than –15.1 with perfect recall. Among only correct span-antecedent pairs, these scores are 11.7 and 7.1, with the same pattern. This indicates that the noise with perfect recall prevents the linker from reliably assigning high coreference scores, even for correct links. The effect of higher coreference scores also shows in that, compared with perfect recall, the perfect precision oracle produces on average larger (4.44 vs. 4.26 entities) and longer-distance (154 vs. 152 tokens spanned) clusters.

We also see this effect by examining the amount of improvement with reduced noise in Table 2. In the anaphoric mention post-hoc oracles, as expected, fixing precision results in fewer extra mention/entity errors and more missing errors, while the perfect recall oracle behaves conversely. However, when re-trained, the perfect precision oracle has much fewer missing entities, even fewer than with perfect recall. This is surprising as the latter considers more candidates. The reason is likely that the linker learns to leverage the absence of noise and reliably assigns high coreference scores. Despite some incorrect links leading to more conflated entities, the many correct ones drastically reduce missing mention/entity errors. On the other hand, the noise in the perfect recall or the original system prevents consistent high scores, resulting in more missing mentions and entities. Hence, the improvement with perfect precision partly stems

[3] Span (boundary) errors are fixed before independently fixing all others. The numbers do not add up to the performance gap due to error type interactions.

	CoNLL-12		PreCo	
	# Op	Op effect	# Op	Op effect
ANA.; P	135.9	0.086	79.2	0.132
ANA.; R	1.9	**0.715**	0.8	**0.709**
ALL; P	34.1	**0.035**	30.6	0.122
ALL; R	115.3	0.002	4.1	**0.143**

Table 4: The number of addition/removal operations needed for the oracle candidates, and the oracle performance increase, in F_1, amortized over each operation. Boldface indicates the higher per-operation effect between perfect precision and recall in each category.

L	30	32	34	36	38	40
F_1	**79.17**	78.76	78.86	79.03	78.88	78.85
λ	0.4	0.45	0.5	0.55	0.6	0.65
F_1	**79.17**	78.99	79.15	78.35	79.05	78.85

Table 5: CoNLL-12 development F_1 with increased max span width L or the number of spans considered per word λ. The first column is the original setting. Boldface indicates the best performance.

from the linker's increased confidence in assigning coreference scores when *not* tasked with ignoring non-mentions (and singletons) in noisy candidates.

The Average Importance of Recall The large improvement from fixing precision may be due to its larger original headroom than recall (Table 1). We compute the number of operations (span addition/removal) needed for each oracle and the average F_1 improvement per operation in Table 4. For anaphoric mentions, recall has 5-8$\times$ the average effect of precision.[4] If we control the number of operations by re-training an anaphoric mention (semi-)perfect precision oracle removing only as many top-scoring extra spans as the number of missing correct spans (rather than removing all extra spans), it gets 79.08 and 85.01 F_1 on CoNLL-12 and PreCo, lower than the perfect recall oracle with 79.65 and 85.22. It is therefore only due to the low-precision high-recall nature of the original detector that precision is more important in aggregate.

Precision-Recall Trade-Off We return to the original question: if we had more compute, is it always beneficial to consider more spans in the detector? From our results, while recall is important, an imprecise detector has substantial adverse effects by increasing the linker's learning burden. Indeed, Table 5 shows that increasing the max span width by up to 33% or the spans considered per word by up to 38% only degrades the performance. As the extra low-scoring spans are mostly noise, we slightly increase recall but more heavily decrease precision, causing more harm than benefit. Hence, besides saving computation, these hyperparameters also balance the precision-recall trade-off. Future work should hence put more emphasis on precision which is often overlooked in end-to-end systems.

5 Difficulties Facing Each Component

5.1 The Detector's Difficulty With Anaphoricity Decisions

Despite its large aggregated improvement, i.e. $\approx$11.7 and 10.5 F_1 for CoNLL-12 and PreCo, perfect anaphoric mention precision requires perfectly distinguishing anaphoric from singleton mentions. These anaphoricity decisions in fact account for most of the improvement, $\approx$10.5 and 6.7 F_1 (Table 3, anaphoric v.s. all mentions perfect precision).[5] However, the detector, as a span classifier, does not explicitly model inter-span anaphoric relationships. To test this architecture's ability to distinguish anaphoric from singleton mentions, we build two span classifiers with the same structure as the detector, supervised with sigmoid loss, that recognize all mentions and anaphoric mentions in PreCo. The former achieves 79.89 classification F_1 while the latter only 54.32, showing the inability of a span classifier to make anaphoricity decisions.

To better understand this difficulty, we define a confusion index as singleton recall divided by anaphoric mention recall. It correlates with the classifier's inability to identify anaphoricity. Ideally, this value should be close to 0, recalling more anaphoric mentions and fewer singletons. A random classifier incapable of distinguishing between the two has an expected confusion index of 1.

The anaphoric mention classifier above has a confusion index of 0.81, showing its inability to make anaphoricity decisions even when explicitly trained with the signal. If we only consider text appearing as both singleton and anaphoric mentions in the same document, demanding contextual reasoning by disregarding obvious anaphoric mentions such as pronouns, the confusion index degrades to 0.997. Hence, the classifier is poor at leveraging self-attentive contextual cues to make anaphoric-

[4] CoNLL-12 with all mentions has a different pattern as we noisily generated singletons in a recall-oriented way.

[5] Chen et al. (2018) observed a similar pattern on an LSTM architecture that less directly receives global information which is important for anaphoricity decisions. We confirm that this still holds on transformers with a larger receptive field.

Error Type (#)	Example
Pronoun (109)	... a cross-sea bridge connecting **Hong Kong, Zhuhai, and Macao** .
	... after **their** return, Macao, and Hong Kong, the two special administrative regions ...
Exact Match (6)	The most important thing about **Disney** is that it is a global brand .
	The subway to **Disney** has already been constructed .
Head Match (11)	**Ten landmark buildings located on Hong Kong Island** reveal themselves ...
	... **those private , er , buildings** , that is , the business community , ah , is willing to ...
Other Match (7)	And **Dr. Andy Henry** notices something else
	Dr. Mann says they 've narrowed it down ...
Semantic Proximity (12)	... **Hong Kong cinema** has nurtured many internationally renowned directors ...
	... memorializing **Hong Kong 's 100 - year film history** .
Others (5)	But [**Paul Kelly**] [**Steve Sodbury**] and Mel Anderson ... had no idea ...

Table 6: Examples of categorized conflated entity errors in the CoNLL-12 development set with a perfect detector. Following past studies (Kummerfeld and Klein, 2013; Joshi et al., 2019), we consider all deictic terms as pronouns. Each example contains two incorrectly linked entities in bold. Square brackets are added to separate mentions.

ity decisions without explicit inter-span relational modeling. In §C we also show the degradation of the confusion index with shorter spans.

Given the importance of anaphoric mention precision (§4), more research in improving anaphoricity decisions in the detector would be fruitful, for example, by more explicitly attending to neighboring spans. Alternatively, as Zhong and Chen (2021) showed the benefit of disentangling the span representations for entity detection and relation extraction in information extraction based on the intuition that they are disparate tasks, one may split the task of anaphoricity decision from mention linking and introduce a separately parameterized anaphoricity module, similarly considering the discrepancy between the two tasks. Recasens et al. (2013); Moosavi and Strube (2016); *inter alia* have pursued similar ideas in the pre-neural era, but it has still not yet been explored with deep models.

5.2 The Linker's Errors

While the detector struggles with anaphoricity decisions, the linker explicitly models anaphoricity by assigning the dummy to extra mentions. It is hence also viable to determine anaphoricity in the linker. Indeed, the current detector would suffice with a stronger linker: in Table 3, the oracle linker gets near-perfect scores with the original mentions (not perfect since the candidates are not gold).[6]

To analyze the remaining non-anaphoricity linker errors, we assume a perfect anaphoric mention detector. Here, conflated entities is the single major error source (last row of Table 2). Table 6 shows 150 manually categorized conflated entities in the CoNLL-12 development set.[7] Suboptimal pronoun resolution is the biggest issue, and the linker also tends to link spans with various degrees of text match or semantic proximity. Within pronoun errors, the most common case is a pronoun linked to an incorrect nominal (in Table 6), occurring 43 times. Sometimes two pronouns, often identical, are incorrectly linked, a case that necessitates better higher-order inference. Third person pronouns with different referents are conflated 29 times. Errors with first or second person pronouns occur 37 times, usually due to speaker switching.

Similar to §5.1, separately parameterizing the linker's encoder may help reduce conflation: intuitively, the span representation for mention detection may promote homogeneity. Meanwhile, the lack of discerning span-internal content for certain error types including pronoun resolution and exact match, combined with current systems' trend to rely on such cues (Lu and Ng, 2020), calls for more focus on improving their contextual reasoning.

6 Conclusion

We analyzed the complex interaction between the mention detector and linker in the mainstream coarse-to-fine coreference system. Using oracle experiments, we showed that, while detector recall is important, higher anaphoric mention precision would lead to dramatically better linker performance, though achieving this is difficult. We also demonstrated that the oracle linker performance is near perfect and that the vast majority of remaining linker errors besides anaphoricity decisions are about pronoun resolution. We hope these discoveries will help future coreference research.

[6] A modified oracle linker that only considers coarse-pruned antecedents (Lee et al., 2018) still gets 96.61 and 98.65 F_1 on CoNLL-12 and PreCo. The small difference compared to considering all antecedents also shows that, with a strong linker, coarse-to-fine pruning has only negligible performance impact while substantially reducing the decision space.

[7] Joshi et al. (2019) and Lu and Ng (2020) conducted similar analyses but we study in a more controlled setting by excluding detector errors and focusing on entity conflation, the largest remaining error source.

References

Saliha Azzam, Kevin Humphreys, and Robert Gaizauskas. 1999. Using coreference chains for text summarization. In *Coreference and Its Applications*.

Hong Chen, Zhenhua Fan, Hao Lu, Alan Yuille, and Shu Rong. 2018. PreCo: A large-scale dataset in preschool vocabulary for coreference resolution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 172–181, Brussels, Belgium. Association for Computational Linguistics. Downloaded from https://preschool-lab.github.io/PreCo.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. 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 5925–5932, Hong Kong, China. Association for Computational Linguistics.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369, Melbourne, Australia. Association for Computational Linguistics.

Zhengbao Jiang, Wei Xu, Jun Araki, and Graham Neubig. 2020. Generalizing natural language analysis through span-relation representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2120–2133, Online. 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.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium. Association for Computational Linguistics.

Jonathan K. Kummerfeld and Dan Klein. 2013. Error-driven analysis of challenges in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 265–277, Seattle, Washington, USA. Association for Computational Linguistics.

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. Association for Computational Linguistics.

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.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

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. 2016. Search space pruning: A simple solution for better coreference resolvers. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1005–1011, San Diego, California. Association for Computational Linguistics.

Haoruo Peng, Daniel Khashabi, and Dan Roth. 2015. Solving hard coreference problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 809–

819, Denver, Colorado. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. 2013. The life and death of discourse entities: Identifying singleton mentions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 627–633, Atlanta, Georgia. Association for Computational Linguistics.

Sanjay Subramanian and Dan Roth. 2019. Improving generalization in coreference resolution via adversarial training. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 192–197, Minneapolis, Minnesota. Association for Computational Linguistics.

Olga Uryupina. 2009. Detecting anaphoricity and antecedenthood for coreference resolution. *Procesamiento del Lenguaje Natural*, 42.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004, San Diego, California. 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.

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.

Rui Zhang, Cícero Nogueira dos Santos, Michihiro Yasunaga, Bing Xiang, and Dragomir Radev. 2018. Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 102–107, Melbourne, Australia. Association for Computational Linguistics.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. 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 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

Zexuan Zhong and Danqi Chen. 2021. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, Online. Association for Computational Linguistics.

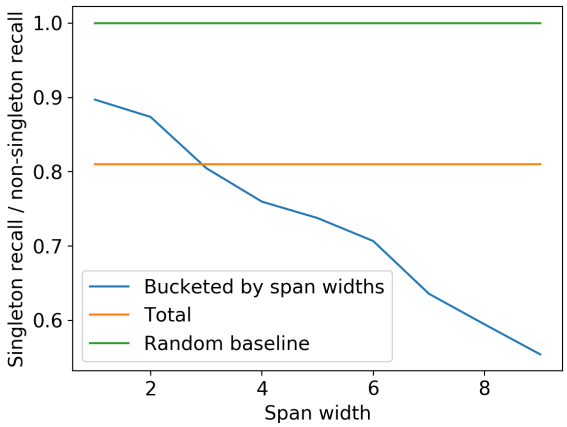

Figure 1: The PreCo anaphoric mention classifier confusion index (§5.1) on spans with different widths.

A Dataset Statistics

We use the English portion of the CoNLL-12 shared task (Pradhan et al., 2012) and the PreCo (Chen et al., 2018) dataset. The former contains 2,802/343/348 training/development/testing documents and the latter has 36.6K training documents and 500 each for development and testing.

B Heuristically Generated CoNLL-12 All Mentions

We heuristically generate the set of all mentions for CoNLL-12 in a recall-oriented manner. We use the gold syntactic information as a proxy and consider the union of all phrases tagged with NP or NML and all words tagged with PRP, PRP$, WP, WDT, WRB, NNP, VB, VBD, VBN, VBG, VBZ, or VBP. This set includes 99.63% anaphoric mentions which constitute 20.89% of this set. We obtain the set of all mentions by merging this set with the non-singleton mentions to ensure all mentions are a superset of anaphoric mentions.

C The Confusion Index's Variation With Span Width

In Figure 1, we plot how the confusion index of the PreCo anaphoric mention classifier (§5.1) changes with span widths. The classifier's inability to make anaphoricity decisions is the most pronounced for short phrases, possibly because these phrases are also more likely to appear as both singleton and anaphoric mentions whose anaphoricity status is especially hard to determine, discussed in §5.1.