CAW-coref: Conjunction-Aware Word-level Coreference Resolution

Karel D'Oosterlinck[1,*], **Semere Kiros Bitew**[1], **Brandon Papineau**[2]
Christopher Potts[2], **Thomas Demeester**[1], **Chris Develder**[1]
[1]Ghent University – imec [2]Stanford University
*karel.doosterlinck@ugent.be

Abstract

State-of-the-art coreference resolutions systems depend on multiple LLM calls per document and are thus prohibitively expensive for many use cases (e.g., information extraction with large corpora). The leading word-level coreference system (`WL-coref`) attains 96.6% of these SOTA systems' performance while being much more efficient. In this work, we identify a routine yet important failure case of `WL-coref`: dealing with conjoined mentions such as *Tom and Mary*. We offer a simple yet effective solution that improves the performance on the OntoNotes test set by 0.9% F1, shrinking the gap between efficient word-level coreference resolution and expensive SOTA approaches by 34.6%. Our Conjunction-Aware Word-level coreference model (`CAW-coref`) and code is available at `https://github.com/KarelDO/wl-coref`.

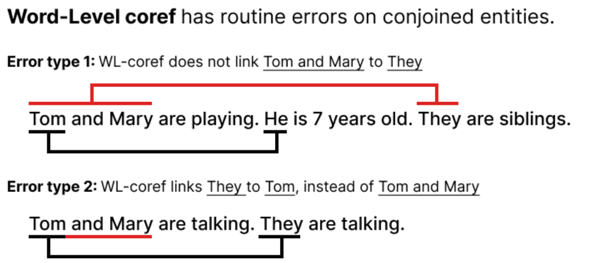

Figure 1: We identify two types of failure cases for `WL-coref` when processing conjoined mentions. Our simple solution, `CAW-coref`, addresses these errors.

1 Introduction

Coreference resolution (or simply *coref*) is the task of clustering mentions in a text, grouping those that refer to the same entity. Coref acts as a fundamental step in many classical NLP pipelines, such as information extraction. Today, however, state-of-the-art (SOTA) coref systems use multiple forward passes of a Large Language model (LLM) *per input document*, making them expensive to train and deploy. This results in limited practical use for classical NLP pipelines, which typically require efficient (and sometimes latency-sensitive) methods.

The most computationally efficient yet competitive neural coref architecture is word-level coref (`WL-coref`; Dobrovolskii, 2021). This method operates by (i) first producing embeddings for each word using one forward pass of a (rather small) LM, then (ii) predicting if pairs of words are coreferent using a lightweight scoring architecture and (iii) finally extracting the spans in the input text associated with these coreferent words. Given a text of n words, this incurs a computational complexity of $O(n^2)$, since the method operates on pairs of words. However, SOTA methods typically perform *multiple* forward passes of a (Large) LM per input document, making them unwieldy for many practical applications. Furthermore, these techniques suffer both from high infrastructure costs and latency-issues associated with these large models.

While significantly less complex, `WL-coref` attains 96.6% of the performance of the current best coreference model (80.7% F1 out of 83.3% F1)[1], as measured on the English split of the OntoNotes dataset (Pradhan et al., 2012). What makes this even more impressive is that `WL-coref` uses one forward pass of a 355M parameter `roberta-large` encoder (Liu et al., 2019), while the state-of-the-art method (Bohnet et al., 2023) uses multiple forward passes of a 13B parameter `mT5-XXL` model (Xue et al., 2021). Thus, `WL-coref` is the go-to architecture for efficiency-sensitive or long-document coref.

In this work, we describe a fundamental weakness of the `WL-coref` model in its original formulation, stemming from how the word-level coref step

[1]Dobrovolskii (2021) reports a performance of 81.0% F1 for `WL-coref` as best performance on the OntoNotes test set. To avoid selecting the best model on the test set, we instead report the test score achieved by our first rerun of `WL-coref` using their code.

was trained. In particular, starting from a dataset that is annotated at the span-level, a word-level dataset is created by using dependency parsing information to select one head-word per span. This causes ambiguity when mentions are conjoined: two spans representing distinct entities can share the same head-word. For example, the span *Tom and Mary* is analyzed as containing three entity mentions (*Tom*, *Mary*, and *Tom and Mary*), and both *Tom and Mary* and *Tom* share the same head-word. When the model at inference time tries to refer both to entity *Tom* and entity *Tom and Mary*, two conflicting links to the span *Tom* are predicted. This causes the model to always drop one of the links, degrading performance (Figure 1).

We resolve this by defining the coordinating conjunction (e.g. *and*, *or*, *plus*) as head-word when faced with these types of mentions, which is a common approach in linguistics (Zoerner III, 1995; Progovac, 1998). Now, the model can learn to systematically link to this conjunction when something is coreferent with *Tom and Mary*, without producing conflicting links. We train a new `WL-coref` model, called Conjunction-Aware Word-level coreference (`CAW-coref`), and find that this simple fix achieves a significant improvement on the OntoNotes test set: the error difference with the state-of-the-art method shrinks by 34.6% (i.e. `CAW-coref` improves the absolute performance of `WL-coref` from 80.7% to 81.6%). Given that this fix incurs no additional model complexity, this gain is an important step forward for efficient coref.

2 Related Work

The main competitive approaches to end-to-end coref can be classified into three broad categories: span-based, word-level, and autoregressive coref.

Span-based coreference Lee et al. (2017) introduce `e2e-coref`, the first end-to-end span-based coref architecture. Starting from word embeddings, the model first predicts which spans are likely a mention. In the second step, coreferent links are predicted between such span-pairs to form coreference clusters. Given a text of n words, this approach incurs $O(n^4)$ computations. Thus, pruning is required to contain the complexity, both for mention prediction and coreference prediction.

Many follow-up works improved upon this architecture by introducing contextualized embeddings (Lee et al., 2018; Kantor and Globerson, 2019), LMs for better span representations (Joshi et al., 2020), ensembling different models for coreference link scoring (`LingMess`; Otmazgin et al., 2023), and distilling the LM backbone for more efficient inference (Otmazgin et al., 2022). Still, the theoretical complexity of these approaches remains $O(n^4)$, requiring pruning and leading to poor scaling on long documents.

Word-level coreference Given an input text, Dobrovolskii (2021) proposes to first predict coreference links between words and subsequently extract the spans surrounding words that are found to be coreferent. This lowers the computational cost of the coref architecture to $O(n^2)$. In turn, less aggressive pruning is needed, which resulted in better performance over conventional span-based techniques.[2] Dobrovolskii (2021) uses one forward pass of a 355M `roberta-large` encoder model to form the contextualized word embeddings needed.

Autoregressive coreference Autoregressive methods iteratively build the coreference structure by running multiple forward passes of an LM backbone. Bohnet et al. (2023) introduce a 13B parameter `mT5-xxl` model called `link-append`: they run multiple forward passes of the LM over increasingly large chunks of the input text and iteratively predict how to grow the coreference structure. This results in the current state-of-the art model on OntoNotes (+2.6% F1 over `WL-coref`). Similarly, Liu et al. (2022) utilize an 11B parameter `Flan-T5-xxl` model (Chung et al., 2022) and predict a sequence of structure-building actions when regressing over the input text (`ASP`). Wu et al. (2020) introduce `corefqa`, formulating coref as a series of question-answering tasks, run multiple forward passes of an LM to build the coreference structure and use extra QA data for augmentation.

In general, the autoregressive methods outperform span-based and word-level coreference, but at great computational cost. All these methods require at least $O(n)$ forward passes of an LM per input document, while span-based or word-level techniques require only one. While some of these computations could be parallelized, running $O(n)$ LM forward passed *per input document* is exceedingly expensive.

Additionally, the `mT5-xxl` and `T0` models used by SOTA methods contain many more parameters

[2] `LingMess` (Otmazgin et al., 2023) is the only span-based method that outperforms `WL-coref`, using a lightweight ensembling technique. This technique could be directly applied to `WL-coref` for potentially a similar performance boost.

compared to the `roberta-large` model used by `WL-coref` (13B and 11B respectively, compared to 355M), making these models less accessible to train and deploy. Liu et al. (2022) show that when using an LM comparable in size to the one used by `WL-coref`, their performance using autoregressive coreference is actually worse. Thus, word-level coreference is the most efficient method in terms of memory requirements and computational scaling.

Error analysis of coreference models Porada et al. (2023) investigate types of errors in recent coref models, including `WL-coref`. Based on the hypothesis that distinct datasets operationalize the task of coreference differently, they perform generalization experiments between multiple datasets and analyze different types of model error. One of their findings suggests that coref for nested mentions is still hard in general.

In this work, we highlight a failure case of `WL-coref`, namely, coreference with conjoined entities (i.e. coordinated noun phrases). We propose and empirically validate a simple yet effective solution.

3 The `WL-coref` model

We briefly summarize the architecture used by Dobrovolskii (2021) and refer to the original publication for a full overview.

Step 1 – Word Representations: First, contextualized word representations are created using one forward pass of an LM backbone and a learned averaging over constituent toMarys.

Step 2 – Word-Level Coreference: To create word-level links, a first *coarse antecedent scoring* is constructed between all pairs of words using a learned bilinear function.

For each word, the top k coarse antecedents are considered in a *fine antecedent scoring step*, using a trained feed forward neural network. The final antecedent scores are given by the sum of the coarse and fine scores. These antecedent scores between pairs of words are used to infer the most likely word-level coreference clustering. The words found to be part of a coreference cluster are passed on to Step 3.

Step 3 – Span Extraction: For each coreferent word, the mention span surrounding it is extracted. This is done using a small feed-forward neural network applied to the contextualized word embeddings, followed by a convolutional layer which predicts probabilities for start and end span boundaries. This step is applied individually for each coreferent word and thus is not directly aware of the global clustering produced in Step 2.

Creating word-level data: To train both steps, Dobrovolskii (2021) uses syntactic information to decompose the span-based OntoNotes dataset into a word-level version and a word-to-span dataset.

The crucial step in this decomposition is selecting one head-word per span. Clearly, these head-words need to be as representative as possible of the entity mentioned in the span, so as to allow the word-level linking to perform well. Additionally, the head-words should be systematically picked so that the span extraction step has an easy time learning to extract the correct span surrounding a coreferent head-word.

Dobrovolskii (2021) picks head-words using dependency parsing information already present in the OntoNotes dataset. Given a span, the method selects the head-word as the word in the span which depends on a word outside of the span. If none or multiple of such words are found, the right-most word of the span is selected as head-word.

4 Failure Modes of `WL-coref`

We describe the two failures cases of `WL-coref` outlined in Figure 1 and propose a simple solution.

Entity Conjunction: `WL-coref` is unable to fully solve routine examples where the conjunction of two or more mentions (e.g. via the use of the coordinating conjunction *and*) forms a new mention in the discourse. Consider the first example from Figure 1: *Tom and Mary are playing. He is 7 years old. They are siblings*. Following how head-words were defined in Dobrovolskii 2021, both the head-word for the mention *Tom and Mary* and the mention *Tom* coincide. At inference time, the word-level coreference step will thus predict both the coreferent links *Tom – He* and *Tom – They*. Since the model does not predict a link *He – They*, one of these two predicted links must be dropped in order to arrive at a consistent clustering. Thus, the model is unable to correctly output both coreferent clusters in this trivial example.

Nested Span Extraction: Given a coreferent head-word, `WL-coref` sometimes struggles to extract the correct span boundaries surrounding this head-word when multiple valid options are possible. Consider the second example from Figure 1: *Tom and Mary are talking. They are talking*. `WL-coref`

	LM		Link	MUC			B^3			CEAF$_{\phi 4}$			Avg.
	calls	params.	compl.	P	R	F1	P	R	F1	P	R	F1	F1
`link-append`	$O(n)$	13B	/	**87.4**	88.3	**87.8**	81.8	**83.4**	**82.6**	79.1	**79.9**	**79.5**	**83.3**
`corefqa`	$O(n^2)$	340M	/	88.6	87.4	88.0	**82.4**	82.0	82.2	**79.9**	78.3	79.1	83.1
`ASP`	$O(n)$	11B	/	86.1	**88.4**	87.2	80.2	83.2	81.7	78.9	78.3	78.6	82.5
`LingMess`	1	355M	$O(n^4)$	85.1	88.1	**86.6**	**78.3**	**82.7**	**80.5**	76.1	78.5	77.3	81.4
`s2e`	1	355M	$O(n^4)$	**85.2**	86.6	85.9	77.9	80.3	79.1	75.4	76.8	76.1	80.3
`CAW` (ours)	1	355M	$O(n^2)$	85.1	**88.2**	**86.6**	77.0	78.0	77.5	**78.0**	**83.2**	**80.6**	**81.6**
`WL`†	1	355M	$O(n^2)$	84.8	87.5	86.1	76.1	76.7	76.6	77.1	82.1	79.5	80.7

Table 1: Results on the OntoNotes 5.0 English test set. Scores calculated with official scorer (Pradhan et al., 2014) or taken from original publication if available. **Avg. F1** is the main metric. We report the amount of LM calls and parameters of the LM used, as well as the coreference linking complexity if applicable. † Dobrovolskii (2021) reports an Avg. F1 of 81.0 as the best `WL-coref` run on the test set, while we report the result of our first run for both `WL-coref` and `CAW-coref`.

correctly predicts the word-level link between *Tom – They*, but fails to extract the span *Tom and Mary* in the subsequent step. This is most likely caused by the span extraction step operating independently on every coreferent head-word: no explicit information about the *Tom – They* link is taMary into account when deciding between *Tom* and *Tom and Mary*, and this decision is thus ambiguous.

Proposed Solution: Both failure modes are rooted in the same fundamental problem: there is no unique one-to-one relation between head-words and spans. This causes issues both when predicting word-level links and when performing span extraction, specifically when dealing with nesting.

We propose to solve this by changing how head-words are defined on conjoined mentions. When creating the word-level training data, we use part-of-speech tags supplied in the OntoNotes dataset to detect if a coordinating conjunction (e.g. *and*, *or*, *plus*) is present in a span. Then we check the relative depth of the conjunction in the dependency parse of the span. If it is less than two steps away from the head-word of the span, it is selected as new head-word. This selects *and* as head-word in the span *Tom and Ann*, but not in the span *David, whose children are called Tom and Ann*. Thus, we have defined a systematic way of picking head-words for conjoined mentions, in a way that they do not conflict with any of the head-words for the nested mentions.

5 Experiments and Results

We use our new word-level dataset to train `CAW-coref`, a new instance of the `WL-coref` architecture. Using our altered notion of head-words, we train and evaluate this model on the English OntoNotes dataset without changing any hyperparameters compared to the default `WL-coref` run. We immediately find an absolute performance increase of 0.9% F1, setting the performance of `CAW-coref` at 81.6% F1. This shrinks the relative gap between efficient coref and expensive SOTA approaches by 34.6%, which is certainly not trivial since gains on OntoNotes have been hard to come by in recent years.

The full breakdown of the results in function of the official evaluation metrics (Vilain et al., 1995; Bagga and Baldwin, 1998; Luo, 2005; Pradhan et al., 2012) is given in Table 1. `CAW-coref` even outperforms `LingMess`, the best span-based method, which uses ensembling to achieve a significant performance boost. Potentially, such an ensembling technique could be applied to further boost `CAW-coref` performance as well.

In total, we found that 1.17% of spans across the English OntoNotes train and development split were such conjoined entities. Supplementary to our empirical analysis, we show the qualitative improvement of `CAW-coref` on a list of simple examples in Appendix A.

6 Conclusion

Neural coreference resolution techniques should be efficient in order to maximize real-word impact. In this work, we outlined two failure cases of the efficient word-level coreference resolution architecture and addressed them with one simple fix. Our new model, Conjunction-Aware Word-level coreference (`CAW-coref`), shrinks the performance gap between efficient and state-of-the-art coreference by 34.6%, and is currently the most performant efficient neural coreference model.

Limitations

There are always more distinct spans than words in a text, thus it is not always possible to uniquely pick a head-word per span. For example, our proposed solution can't fully handle sequential conjunctions such as *Tom and Mary and David*, since this span contains only 5 words but 6 mentions: *Tom*, *Tom and Mary*, *Mary*, *Mary and David*, *David*, and *Tom and Mary and David*. Luckily, we did not observe any such dense references in the dataset.

Our procedure of selecting a new head-word for conjunctions relies on syntactic information in the form of part-of-speech tags and dependency parses. OntoNotes features several instances where conjunctions are formed using commas or hyphens, such as in the span *Tom , Mary* or *Tom - Mary*. Here, the comma and hyphen should take on the role as head-word of the conjunction, but this is much harder to detect using the syntactic information present.

Future work could focus on resolving both these issues to further boost the performance of efficient Conjunction-Aware Word-level coreference resolution.

Acknowledgements

We are grateful to our anonymous reviewers for their meticulous reading and valuable comments. Karel D'Oosterlinck is funded by an FWO Fundamental Research PhD Fellowship (11632223N).

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains.

Bernd Bohnet, Chris Alberti, and Michael Collins. 2023. Coreference resolution through a seq2seq transition-based system. *Transactions of the Association for Computational Linguistics*, 11:212–226.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

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.

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.

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.

Tianyu Liu, Yuchen Eleanor Jiang, Nicholas Monath, Ryan Cotterell, and Mrinmaya Sachan. 2022. Autoregressive structured prediction with language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 993–1005.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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.

Shon Otmazgin, Arie Cattan, and Yoav Goldberg. 2022. F-coref: Fast, accurate and easy to use coreference resolution. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 48–56, Taipei, Taiwan. Association for Computational Linguistics.

Shon Otmazgin, Arie Cattan, and Yoav Goldberg. 2023. Lingmess: Linguistically informed multi expert scorers for coreference resolution. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2744–2752.

Ian Porada, Alexandra Olteanu, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2023. Investigating failures to generalize for coreference resolution models. *arXiv preprint arXiv:2303.09092*.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–35, Baltimore, Maryland. 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, Jeju Island, Korea. Association for Computational Linguistics.

Ljiljana Progovac. 1998. Structure for coordination. *Glot international*, 3(7):3–6.

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*.

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.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Cyril Edward Zoerner III. 1995. *Coordination: The syntax of &P*. University of California, Irvine.

Model	Step	Prediction	Correct
WL-coref	word	Tom and Anna are talking. They are talking.	Yes
WL-coref	span	Tom and Anna are talking. They are talking.	No
CAW-coref	word	Tom and Anna are talking. They are talking.	Yes
CAW-coref	span	Tom and Anna are talking. They are talking.	Yes
WL-coref	word	My friend David and my dad Bert are talking. They are talking.	No
WL-coref	span	My friend David and my dad Bert are talking. They are talking.	No
CAW-coref	word	My friend David and my dad Bert are talking. They are talking.	Yes
CAW-coref	span	My friend David and my dad Bert are talking. They are talking.	Yes
WL-coref	word	The Guardian and The Chronicle had a secret meeting . Both newspapers are on thin ice .	No
WL-coref	span	The Guardian and The Chronicle had a secret meeting . Both newspapers are on thin ice .	No
CAW-coref	word	The Guardian and The Chronicle had a secret meeting . Both newspapers are on thin ice .	Yes
CAW-coref	span	The Guardian and The Chronicle had a secret meeting . Both newspapers are on thin ice .	Yes

Table 2: Three hand-crafted examples and their word-level and span-level predictions for `WL-coref` and `CAW-coref`. Coreferent predictions are indicated with a colored box, where each unique entity has the same color. Predictions are considered correct or not correct for their respective step in the word-level pipeline.

A Qualitative Examples

Three qualitative examples comparing `WL-coref` and `CAW-coref` with the word-level and span-level predictions are given in Table 2.