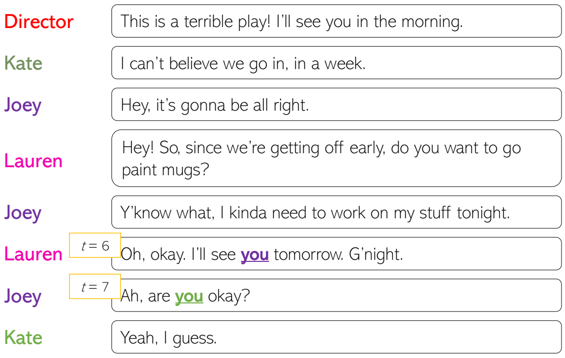
Online Neural Coreference Resolution with Rollback

Patrick Xia and **Benjamin Van Durme**
Human Language Technology Center of Excellence
Johns Hopkins University
{paxia, vandurme}@cs.jhu.edu

Abstract

Humans process natural language online, whether reading a document or participating in multiparty dialogue. Recent advances in neural coreference resolution have focused on offline approaches that assume the full communication history as input. This is neither realistic nor sufficient if we wish to support dialogue understanding in real-time. We benchmark two existing, offline, models and highlight their shortcomings in the online setting. We then modify these models to perform online inference and introduce *rollback*: a short-term mechanism to correct mistakes. We demonstrate across five English datasets the effectiveness of this approach against an offline and a naive online model in terms of latency, final document-level coreference F1, and average running F1.



Director: This is a terrible play! I'll see you in the morning.
Kate: I can't believe we go in, in a week.
Joey: Hey, it's gonna be all right.
Lauren: Hey! So, since we're getting off early, do you want to go paint mugs?
Joey: Y'know what, I kinda need to work on my stuff tonight.
Lauren: Oh, okay. I'll see **you** tomorrow. G'night.
Joey: Ah, are **you** okay?
Kate: Yeah, I guess.

Figure 1: In this scene from *Friends*, viewers can deduce who "you" refers to at $t = 6$, and we want coreference models to be similarly capable. At $t = 7$, viewers may need more context, such as the identity of the next speaker, to be certain of who "you" refers to. Absent that context for a text-based model, its predictions will be incorrect. Our proposed *rollback* is a cheap and local revision mechanism that corrects these type of mistakes.

1 Introduction

In environments like multiparty spoken dialogue and social media streams, text in the form of tokens and sentences are available in (near) real-time. To promptly make use of this data, NLP systems often need to process text before additional tokens or sentences are available. For example, this could enable interruptions with a response or a clarification question (Boyle et al., 1994; Li et al., 2017), make decisions during a social media stream (Mathioudakis and Koudas, 2010), or recognize and translate speech live (Oda et al., 2014; Ma et al., 2020). While some language technologies operate incrementally in the *online* setting, many document-level understanding models and tasks do not.

A core task in language understanding is resolving references. Recent work has made significant progress on improving accuracy for single documents (Lee et al., 2017; Wu et al., 2020) and in the cross-document setting (Caciularu et al., 2021). However, this focus on document-level resolution makes use of global higher order inference and document-level encodings. As interest in coreference resolution is shifting back towards dialogue (Khosla et al., 2021), the *offline* setting is inconsistent with how dialogue is found in the real world. Now equipped with neural models and large-scale data, we revisit the *online* coreference resolution setting (Stoness et al., 2004; Schlangen et al., 2009).[1]

In this work, we are motivated by the human ability to resolve references *without looking into the future* (Figure 1). We simulate the online setting for two offline models (Xu and Choi, 2020; Xia et al., 2020) by making full predictions after each sentence and masking the future context. This either leads to significantly increased latency or lowered accuracy. We then modify the latter model to properly perform online inference and show that while accuracy does drop relative to the offline baselines, the latency is substantially lower. Finally, we propose *rollback*, a backtracking method which allows

[1] Xu and Choi (2022) recently explore the online setting in contemporaneous work.

Proceedings of the Fifth Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2022) , pages 13–21
Gyeongju, Republic of Korea, October 16–17, 2022.

the model to correct recently made decisions. On several coreference datasets, we show that this can recover performance comparable to that of the offline model with the latency of online models.

2 Task: Online Coreference Resolution

In offline (single doc) coreference resolution, the input is a document D, and the output is a set of clusters (or chains) of text mentions, $\mathcal{C} = \{C_1, ..., C_n\}$ such that any two mentions in a given C_i corefer. Evaluation can be performed at the document level, $S(\mathcal{C}_{\text{pred}}, \mathcal{C}_{\text{gold}})$, by comparing the predicted clusters to the gold reference clusters with an average of three corpus-level metrics, MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and CEAF$_{\phi_4}$ (Luo, 2005), for the accuracy of mentions, links, and clusters. When each metric is instead computed at the corpus level instead before averaging, we refer to this as *final F1* (identical to CoNLL 2012 F1).

In the sentence-level online setting, $D = [X_1, X_2, ..., X_T]$ is a stream of sentences or utterances. After time t, we predict clusters $\mathcal{C}_{\text{pred},t} = \{C_{1,t}, ..., C_{n,t}\}$ conditioned on only $[X_1, ...X_t]$. For the reference clusters $\mathcal{C}_{\text{gold},t}$, we restrict clusters in $\mathcal{C}_{\text{gold},t}$ to contain only mentions up to sentence X_t. This may lead to empty clusters which are ignored when calculating the score.[2] To evaluate, we propose additionally using *running F1* for each document:

$$S_{\text{running}}(\mathcal{C}_{\text{pred}}, \mathcal{C}_{\text{gold}}) = \sum_{t=1}^{T} \frac{1}{T} S(\mathcal{C}_{\text{pred},t}, \mathcal{C}_{\text{gold},t}).$$

These document-level scores are subsequently averaged across the corpus (macro-average), in contrast to the already corpus-level metrics of *final F1*.

We are not the first to observe that references should be resolvable without future context. Prior work (Stoness et al., 2004; Schlangen et al., 2009; Poesio and Rieser, 2011) has also emphasized the importance of incremental (online) prediction of reference, especially in the context of dialogue. Since most models at that time already operated at the sentence level, their work is at the token-level granularity. Our work does not go as far; our goal is to first rein back *document* level neural models to the sentence level, which is still appropriate in applications where full utterances are available.

Dataset	Training	Dev	Test	Avg. sents
OntoNotes$^{\text{all}}$	2,802	343	348	26.8
OntoNotes$^{\text{conv.}}$	393	75	71	54.9
OntoNotes$^{\text{text.}}$	2,409	268	277	22.2
CI	987	122	192	19.0
LitBank	80	10	10	84.4
QBCoref	240	80	80	4.7

Table 1: Number of documents in each split for each corpus considered in this work. Avg. sents refers to number of sentences per document in the training set

Finally, we would like to compare the latency of different systems. Unlike token-level work in speech (Zhang et al., 2016) or translation (Gu et al., 2017), we are primarily interested in sentences, and we do not have readily available timestamps. Furthermore, modern models can process a single sentence in under a second, while sentences take substantially longer to be spoken or typed. Therefore, we mainly report document-level latency, which is the *wait time* between the end of the document and production of predictions. We revisit and discuss sentence-level latency in Section 4.4.

3 Method

3.1 Datasets

We select several coreference datasets to study, detailed in Table 1, that will let us analyze a variety of domains. We split the CoNLL 2012 Shared Task (OntoNotes) (Pradhan et al., 2013) into the conversational (telephone and broadcast conversations) and nonconversational text (newswire, newsgroups, broadcast news, weblogs, religious texts) genres. Character Identification (CI) (Zhou and Choi, 2018) consists of transcripts from the TV show *Friends* and is another source of social and informal conversations. LitBank (Bamman et al., 2020) is a collection of long excerpts from literature, which allows us to study latency scaling. Finally, QBCoref (Guha et al., 2015) is a collection of trivia questions where players are expected to interrupt with the answer, which is an example of a task needing a fast NLU model.

3.2 Models

We use Xu and Choi (2020) and Xia et al. (2020) as our offline baselines. We then modify the inference algorithm of the latter for our online experiments.[3]

[2] Singletons may also be ignored depending on the dataset.

[3] Code is available at `https://github.com/pitrack/incremental-coref`.

C2F (Xu and Choi, 2020) is a reimplementation of the coarse-to-fine coreference model (Lee et al., 2018) which detects mention spans in the entire document, scores them with each other, and finds the most likely antecedent for each span. It then uses higher order decoding strategies to promote pairwise consistency within a cluster. In this work, we do not use these decoding strategies as they are slower and only improve performance slightly. We do, however, use the extension to the training loss that accommodates singletons (Xu and Choi, 2021).

ICOREF (Xia et al., 2020) is a memory-efficient incremental coreference resolution model, itself a variant of the C2F model. They achieve this by segmenting the document into pieces that fit into a single SpanBERT (Joshi et al., 2020) window, incrementally processing each segment, and saving the set of found entity clusters after each step. Within each segment, they detect mention spans, find each span's most likely entity cluster, merge it (or form a new cluster), and update that cluster's embedding. After each text segment, the predictions for that segment are committed. This hard decision foregoes any higher-order decoding strategies, but this locality offered is exactly what we wish to extend in the sentence-level online setting.

Naive online C2F is a baseline where C2F is used to make full predictions after every sentence. For a document with n sentences, this costs n calls to the full C2F model, and effectively acts as an upper limit on model performance.

Online ICOREF For the online models, we choose to modify the inference process in ICOREF. This is because ICOREF already processes the document incrementally and it also foregoes global inference across all clusters. Like prior models, ICOREF encodes a variable number of sentences per encoder forward pass, and each sentence would have access to future contexts. To make this fully online, we modify the algorithm by segmenting the text by sentences instead of by tokens. Thus, instead of making predictions every fixed number of tokens (e.g. 512), the model makes predictions every u sentences. Setting $u = 1$ would make an online model at the sentence level.

Online ICOREF with rollback A drawback of both ICOREF and online modeling in general is the inability to correct mistakes in light of future context. We also introduce "rollback," which is run every r sentences (Algorithm 1). This process reverts all predictions made in the previous r sentence-window and remakes them all, batch-mode, with the full (r-sentence) context. The trade-off of increasing r is that the intermediate prediction quality can suffer, while decreasing r incurs additional latency.

Algorithm 1 Online coreference with rollback

```
Input: Sentences $S = s_1, s_2, ...$; update frequency $u$; rollback frequency $r$; initial clusters $\mathcal{C}_0 = \emptyset$.
for $s_t \in S$ do
    if $t \equiv 0 \pmod{ur}$ then
        $\mathcal{C}_{t-ur+1} = \text{REVERT}(\mathcal{C}_{t-1})$
        $\mathcal{C}_t = \text{ICOREF}(S[t-ur+1:t], \mathcal{C}_{t-ur+1})$
    else if $t \equiv 0 \pmod{u}$ then
        $\mathcal{C}_t = \text{ICOREF}(S[t-u+1:t], \mathcal{C}_{t-1})$
    else
        $\mathcal{C}_t = \mathcal{C}_{t-1}$
    yield $\mathcal{C}_t$
```

Δ Final F1	C2F		ICOREF	
Masked Training?	No	Yes	No	Yes
OntoNotes$^{\text{conv}}$	-7.8	-1.8	-8.0	-7.6
OntoNotes$^{\text{text}}$	-6.0	-0.3	-8.0	-6.9
LitBank	-5.3	-1.9	-5.1	-5.4
QBCoref	-4.9	-0.5	-1.1	-2.7
CI	-5.5	-1.0	-11.0	-9.6

Table 2: We train a model with and without sentence-level causal attention masks. We then report the difference in F1 between inference with and without this mask in the offline setting. Full numbers in Appendix C.

4 Experiments and Results

We first show that current models rely on future context, which is not readily available in the online setting. We demonstrate the effectiveness of online models under latency and average running F1. In particular, we analyze the benefits of rollback. Finally, we verify that for reasonable input stream speeds, online approaches are indeed appropriate.

4.1 Masking the future

We first investigate the reliance of the two baseline (offline) models, C2F and ICOREF, on future context. As shown in Figure 1, models often use future contexts to make predictions such as linking "you" with the next speaker. For each model, we consider applying a sentence-level causal mask in the encoder and remove any global decoding algorithms. The causal mask restricts each token's attention only to other tokens in its sentence or a previous one. With this mask at inference, we find that with

	naive online C2F			ICoref			Online ICoref			+ rollback		
Metric	Run. F1	Fin. F1	wt (ms)	Run. F1	Fin. F1	wt (ms)	Run. F1	Fin. F1	wt (ms)	Run. F1	Fin. F1	wt (ms)
OntoNotes$^{\text{conv}}$	79.2	77.0	237.8	24.9	76.2	319.3	74.8	72.7	52.0	76.6	75.2	79.0
OntoNotes$^{\text{text}}$	82.3	80.6	195.2	28.9	80.5	223.8	77.8	77.4	62.1	79.1	79.9	87.9
LitBank	73.8	72.2	807.4	54.5	72.7	173.3	71.9	70.6	73.5	72.6	71.3	93.7
QBCoref	76.6	70.5	107.9	15.6	71.9	82.3	72.5	71.1	45.8	72.7	71.6	54.9
CI	74.7	73.0	137.5	14.2	71.9	227.8	65.1	66.7	47.3	67.3	70.1	59.3

Table 3: Final F1, running F1, and wait time for each datasets and four inference algorithms. Our proposed rollback mechanism offers a strong compromise with higher F1s and comparable wait times vs. the fastest online models, and a final F1 comparable to offline ICoref. Naive online C2F is the strongest method, but also the slowest.

Dataset	#Edits	Ment.	New	Existing
LitBank	453	12.1, 9.3	12.6, 10.4	27.2, 6.0
QBCoref	145	20.0, 8.3	16.6, 13.1	16.6, 7.6
CI	429	4.9, 4.4	17.0, 5.6	27.0, 13.3

Table 4: We classify the edits made in each dev set via rollback: **Ment**ion detection errors, missed **New** clusters, and incorrect links to **Existing** clusters. We report the percentage of (wrong→right, right→wrong) edits. The unreported fraction of edits are wrong→wrong. We omit OntoNotes because that dataset does not include singleton clusters, making this type of analysis difficult.

both models, performance drops considerably (Table 2). However, by training with the causal mask, the C2F model recovers from these drops in the masked setting. This suggests that coreference resolution models can be retrained to make better use of previous context and rely less on “easy” future signals. This finding is also quite promising for future investigation into *training* methods.

On the other hand, masked training does not affect the performance of the ICoref model. Nonetheless, the incremental nature of ICoref and ability to predict singletons is more amenable to extension to an online setting, and so we proceed with ICoref without masking.

4.2 Online inference strategies

To properly evaluate online performance (as opposed to only simulating masking the future), we apply the modifications to ICoref described in Section 3.2 and compare the running F1, final F1, and wait time. By increasing update sizes, u, we can interpolate between an online model ($u = 1$) and the unmasked offline ICoref model (where u is the encoder window size). This “hybrid” mode trades off wait time for F1, as increasing u leads to longer wait times but better performance. In addition, we find that changing the rollback frequency does not correlate with wait time because larger updates are both costlier and rarer. So, we choose the best r based on each dev set.

Table 3 shows that the online models are faster than the offline ICoref model and do better on running F1, but worse on final F1. Online with rollback is usually the best approach, as it achieves high F1 scores across all datasets, while it also has short wait times. Naive online C2F performs well on F1, but it is substantially slower on especially short or long documents.

The small margin on QBCoref could be explained by the fact that the forward pass for online ICoref is equal to that of a causally masked offline model and Table 2 shows that the gap between a masked and unmasked model is small.

4.3 Error correction with rollback

In Table 4, we calculate the number of predictions that are changed with rollback. In general, more edits are corrections (wrong→right) than errors (right→wrong), which demonstrates the effectiveness of rollback. For all three datasets, many of the corrections made address correctly assigning spans to existing clusters, such as the "you" in Figure 1. In QBCoref, many corrections are un-predicting a non-mention, while in CI, many corrections are correctly predicting new starts of entity clusters.

4.4 Latency analysis

In this work, we assume that each sentence arrives after all computation has been completed for the previous sentence, which motivates our use of wait time as a metric. However, this assumption may not always be true in situations where utterances are highly frequent or short, like in online chat rooms.

To verify this empirically, we run simulations to find the token arrival speeds for which offline and online models have equivalent *sentence* latency (details in Appendix E). For all datasets, we find that this point is at over 200 words per second (wps). Additionally, if the stream is slower than 20 wps, there is never a "delay" caused by processing a sentence. This is substantially faster than the speaking (Yuan et al., 2006) and reading (Brysbaert, 2019) rates of around 3-5 wps. Therefore, sentence-level predictions are being made faster than tokens are produced, which validates our metric of wait time in this work. This may not extend to some settings with high arrival rates, like livestream comments.

5 Conclusion

We look at reining back document-level models for neural coreference resolution to the utterance level by proposing a shift towards online inference. We propose a model with the capability for making predictions online, after every sentence. This leads to lower latency than a corresponding offline model, and maintains a consistently high running F1 after each sentence. To edit predictions made without future context, we introduce a rollback mechanism which reverts and corrects recently made predictions, bringing the F1 closer to that of the offline model while maintaining its ability to make online predictions with low latency.

Future work may consider extensions to this approach by handling online processing at the word-level, revisiting the scenario considered by Schlangen et al. (2009).

Acknowledgements

We would like to thank Xutai Ma, Mahsa Yarmohammadi, and Boyuan Zheng for helpful feedback and discussion on this work. In addition, we appreciate the insightful comments and questions from the anonymous reviewers.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566.

David Bamman, Olivia Lewke, and Anya Mansoor. 2020. An annotated dataset of coreference in English literature. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 44–54, Marseille, France. European Language Resources Association.

Elizabeth A Boyle, Anne H Anderson, and Alison Newlands. 1994. The effects of visibility on dialogue and performance in a cooperative problem solving task. *Language and speech*, 37(1):1–20.

Marc Brysbaert. 2019. How many words do we read per minute? a review and meta-analysis of reading rate. *Journal of memory and language*, 109:104047.

Avi Caciularu, Arman Cohan, Iz Beltagy, Matthew Peters, Arie Cattan, and Ido Dagan. 2021. CDLM: Cross-document language modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2648–2662, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor O.K. Li. 2017. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1053–1062, Valencia, Spain. Association for Computational Linguistics.

Anupam Guha, Mohit Iyyer, Danny Bouman, and Jordan Boyd-Graber. 2015. Removing the training wheels: A coreference dataset that entertains humans and challenges computers. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1108–1118, Denver, Colorado. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, 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.

Sopan Khosla, Juntao Yu, Ramesh Manuvinakurike, Vincent Ng, Massimo Poesio, Michael Strube, and Carolyn Rosé. 2021. The CODI-CRAC 2021 shared task on anaphora, bridging, and discourse deixis in dialogue. In *Proceedings of the CODI-CRAC 2021 Shared Task on Anaphora, Bridging, and Discourse Deixis in Dialogue*, pages 1–15, Punta Cana, Dominican Republic. 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.

Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2017. Learning through dialogue interactions by asking questions. In *ICLR*.

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.

Xutai Ma, Juan Pino, and Philipp Koehn. 2020. SimulMT to SimulST: Adapting simultaneous text translation to end-to-end simultaneous speech translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 582–587, Suzhou, China. Association for Computational Linguistics.

Michael Mathioudakis and Nick Koudas. 2010. Twittermonitor: Trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, page 1155–1158, New York, NY, USA. Association for Computing Machinery.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2014. Optimizing segmentation strategies for simultaneous speech translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 551–556, Baltimore, Maryland. Association for Computational Linguistics.

Massimo Poesio and Hannes Rieser. 2011. An incremental model of anaphora and reference resolution based on resource situations. *Dialogue Discourse*, 2:235–277.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of CoNLL*.

David Schlangen, Timo Baumann, and Michaela Atterer. 2009. Incremental reference resolution: The task, metrics for evaluation, and a Bayesian filtering model that is sensitive to disfluencies. In *Proceedings of the SIGDIAL 2009 Conference*, pages 30–37, London, UK. Association for Computational Linguistics.

Scott C. Stoness, Joel Tetreault, and James Allen. 2004. Incremental parsing with reference interaction. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 18–25, Barcelona, Spain. Association for Computational Linguistics.

Shubham Toshniwal, Patrick Xia, Sam Wiseman, Karen Livescu, and Kevin Gimpel. 2021. On generalization in coreference resolution. In *Proceedings of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 111–120, Punta Cana, Dominican Republic. 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*.

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.

Patrick Xia and Benjamin Van Durme. 2021. Moving on from OntoNotes: Coreference resolution model transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5241–5256, Online and Punta Cana, Dominican Republic. 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.

Liyan Xu and Jinho D. Choi. 2021. Adapted end-to-end coreference resolution system for anaphoric identities in dialogues. In *Proceedings of the CODI-CRAC 2021 Shared Task on Anaphora, Bridging, and Discourse Deixis in Dialogue*, pages 55–62, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Liyan Xu and Jinho D. Choi. 2022. Online coreference resolution for dialogue processing: Improving mention-linking on real-time conversations. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 341–347, Seattle, Washington. Association for Computational Linguistics.

Jiahong Yuan, Mark Liberman, and Christopher Cieri. 2006. Towards an integrated understanding of speaking rate in conversation. In *Ninth International Conference on Spoken Language Processing*.

Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yao, Sanjeev Khudanpur, and James Glass. 2016. Highway long short-term memory rnns for distant speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5755–5759. IEEE.

Ethan Zhou and Jinho D. Choi. 2018. They exist! introducing plural mentions to coreference resolution and entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 24–34, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

A Experimental Details

A.1 Datasets Preprocessing

We use the same preprocessing as Joshi et al. (2019) for OntoNotes, Xia and Van Durme (2021) for LitBank (first fold) and QBCoref (first fold), and Toshniwal et al. (2021) for CI. For the genre split in OntoNotes, we split the full dataset into a conversational and text-based component. Some weblog documents are conversations on message boards. We maintain this split because they are less conversational than spoken dialogue, and it is consistent with the split originally used in by ICOREF. While OntoNotes does have non-English splits, we only study English data in this work. To our knowledge, the datasets and codebases were released intended to advance research in coreference resolution, which is aligned with the focus of this work.

Since ICOREF does not readily take speaker embeddings, we augment the underlying text of CI with speakers by prepending each utterance with the name of the speaker(s), following the strategy outlined by Wu et al. (2020), and we only filter out these mentions before evaluation. We note that there could be other ways of representing the speakers, especially in plural situations, which we do not explore as it is beyond the scope of the work. While this follows the same preprocessing as Toshniwal et al. (2021), we do not do this for C2F, as this model uses the speakers as a feature. We do not evaluate CI following the metrics outlined in Zhou and Choi (2018) as we are primarily interested in exploring online coreference by using the dialogue and conversational nature of the dataset and not in the plural mentions and multiparty aspect.

A.2 Hyperparameters

We maintain all the default hyperparameters for both the C2F model[4] and ICOREF model.[5] For C2F, we train with and without mention detection loss (coefficient=1), depending on the dataset. At inference, we would also include positive scoring mentions in the predicted clusters. In addition, we follow the previous findings on continued training (Gururangan et al., 2020; Xia and Van Durme, 2021) by continuing training from the publicly released OntoNotes checkpoints of each model. We train each model once. Again, the goal of our short paper is to highlight online coreference resolution, specifically, online *inference*.

To that end, we explore several values of $u \in [1, 2, 3, 4, 5, 6, 7, 8]$ and $r \in [2, 4, 5, 6, 8, \text{no rollback}]$ for each of the datasets. We plot u in Figure 2 to interpolate between the online and offline models. We select $r = 4$ for QBCoref, $r = 6$ for LitBank, and $r = 8$ for the other splits. Furthermore, following the findings in Section 4.1, we train all models with and without the causal mask. Models without the mask performs better.

For each test set and model (i.e. point in Figure 2), we run inference three times and take the *minimum* time rather than the average. We use minimum because in rare cases, one of the runs would be significantly slower, which would disproportionately affect the average. Overall, the mean difference between the max and min wait time across all datasets is around 10.5ms, or 12% relative to the min wait time, and the median is 5.8ms.

A.3 Computing Revisions

To compute revisions due to rollback in Section 4.3, we split each mention identified by the model either before or after rollback based on its gold reference antecedent: not a mention, first mention of a cluster, or part of another cluster. We count the number of revisions for the first two classes. For the third, we consider a cluster link correct if the

[4] https://github.com/lxucs/coref-hoi

[5] https://github.com/pitrack/incremental-coref/

Δ Final F1	C2F				ICoref			
Masked Training?	No		Yes		No		Yes	
Masked Inference?	Yes	No	Yes	No	Yes	No	Yes	No
OntoNotes^{conv}	69.2	77.0	75.0	76.7	68.2	76.2	68.4	76.0
OntoNotes^{text}	74.7	80.6	79.9	80.2	72.5	80.5	73.4	80.3
LitBank	66.9	72.2	68.8	70.7	67.6	72.7	67.5	72.9
QBCoref	64.9	69.8	70.0	70.5	70.8	71.9	69.7	72.5
CI	67.6	73.0	71.8	72.8	60.9	71.9	61.2	70.9

Table 5: This is the full version of Table 2, on the test set. Each entry instead shows the score with mask and the score without mask instead of the difference

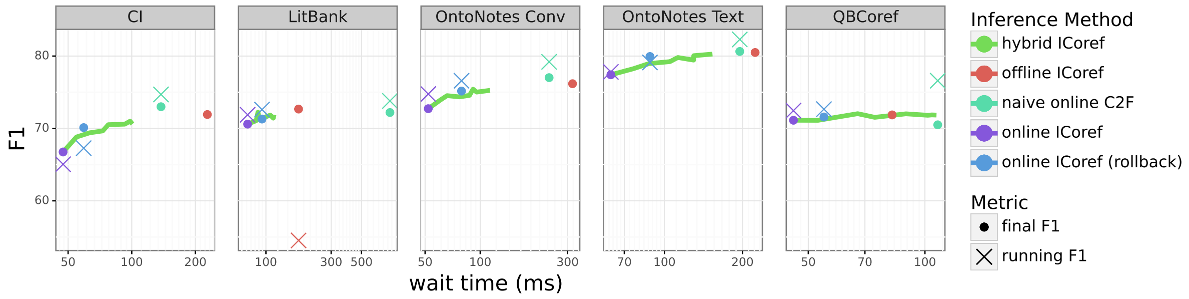


Figure 2: We plot the average wait time against the final F1 (test) and the running F1 (×) for select models. By varying the update frequency, we interpolate between online and offline ICoref models in both final F1 and wait time.

majority of the predicted cluster overlaps with the reference cluster.

A.4 Compute

We run all experiments on a single NVIDIA RTX Quadro 6000 GPU. Training each model completes in under 24 hours, with some datasets like QBCoref taking significantly less times (under an hour). Inference runs in 1-5 minutes per trial. Because our focus was not on training (we trained each model only once and we leveraged continued training), we estimate we use around 15 GPU-days for all results presented in this paper, and not substantially (at most 3x) more than that in the development of this work. Each model is dominated by the size of SpanBERT-large (334M). C2F models have 381M parameters and ICoref has 373M.

B Usage

Like any improvements to information extraction or natural language understanding technologies, malicious users can more easily automate harmful applications (e.g. illegal web scraping). For this work in particular, introducing an online coreference resolution model could make such applications even faster and shift the paradigm further towards harmful (algorithmically) online applications. Nonetheless, these coreference resolution models themselves are not a complete technology, and so the harms of this work are minimal. Both of the baseline models we use in this work and the subsequently released code are licensed under Apache 2.0.

C Masked Training and Inference

Table 5 is a more complete version of Table 2 from Section 4.1.

D Visual comparison of strategies

We can also visualize Table 3 in Figure 2, which shows several inference procedures. This figure more clearly shows that by modifying the rollback frequency, a hybrid inference method can be chosen to favor a purely online approach or a slower, offline approach.

E Latency

To compute sentence-level latency, we assume each (sub)token arrives uniformly at a specified rate. When the last token of a sentence arrives, if the model decides to process the preceeding chunk, we simulate running inference over the previous sentence(s). In parallel, we assume tokens continue arriving.

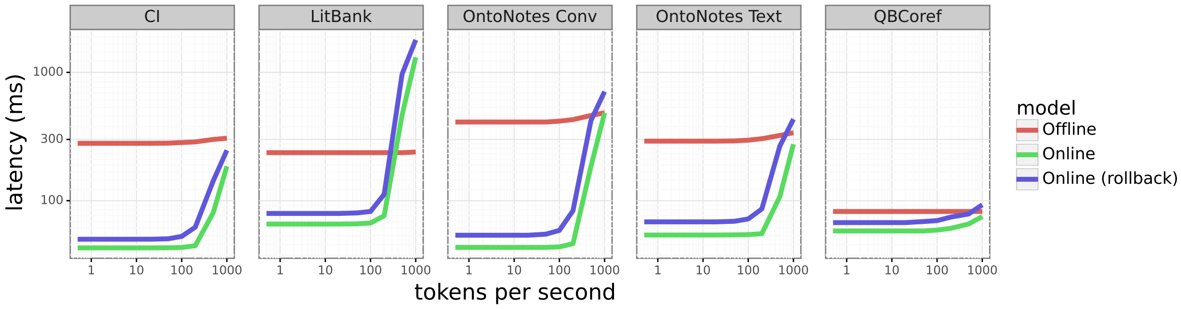


Figure 3: Simulated mean sentence-level latency given different token arrival rates.

We compute the latency between the end of *each sentence* and when the predictions *for that sentence* are produced by the simulated model. Since ICOREF is sequential, if the model is due to process a segment before the previous one is completed, the next segment is blocked until the previous one is complete.

We run inference once to obtain the size of the job for each of these segments, and then simulate sentence-level latency with different rates. We do this for just the online and offline ICOREF models, as the goal is to gain some intuition over token arrival rates and these were usually the fastest and slowest. The results are plotted in Figure 3