On Generalization in Coreference Resolution

Shubham Toshniwal[1]*, Patrick Xia[2]*, Sam Wiseman[3], Karen Livescu[1], Kevin Gimpel[1]
[1]Toyota Technological Institute at Chicago
[2]Johns Hopkins University
[3]Duke University

{shtoshni, klivescu, kgimpel}@ttic.edu, paxia@cs.jhu.edu, swiseman@cs.duke.edu

Abstract

While coreference resolution is defined independently of dataset domain, most models for performing coreference resolution do not transfer well to unseen domains. We consolidate a set of 8 coreference resolution datasets targeting different domains to evaluate the off-the-shelf performance of models. We then mix three datasets for training; even though their domain, annotation guidelines, and metadata differ, we propose a method for jointly training a single model on this heterogeneous data mixture by using data augmentation to account for annotation differences and sampling to balance the data quantities. We find that in a zero-shot setting, models trained on a single dataset transfer poorly while joint training yields improved overall performance, leading to better generalization in coreference resolution models. This work contributes a new benchmark for robust coreference resolution and multiple new state-of-the-art results.[1]

1 Introduction

Coreference resolution is a core component of the NLP pipeline, as determining which mentions in text refer to the same entity is used for a wide variety of downstream tasks like knowledge extraction (Li et al., 2020), question answering (Dhingra et al., 2018), and dialog systems (Gao et al., 2019). As these tasks span many domains, we need coreference models to generalize well.

Meanwhile, models for coreference resolution have improved due to neural architectures with millions of parameters and the emergence of pretrained encoders. However, model generalization across domains has always been a challenge (Yang et al., 2012; Zhao and Ng, 2014; Poot and van Cranenburgh, 2020; Aktaş et al., 2020). Since these models are usually engineered for a single dataset, they capture idiosyncrasies inherent in that dataset. As an example, OntoNotes (Weischedel et al., 2013), a widely-used general-purpose dataset, provides metadata, like the document genre and speaker information. However, this assumption cannot be made more broadly, especially if the input is raw text (Wiseman et al., 2016).

Furthermore, while there are datasets aimed at capturing a broad set of genres (Weischedel et al., 2013; Poesio et al., 2018; Zhu et al., 2021), they are not mutually compatible due to differences in annotation guidelines. For example, some datasets do not annotate singleton clusters (clusters with a single mention). Ideally, we would like a coreference model to be robust to all the standard datasets. In this work, we consolidate 8 datasets spanning multiple domains, document lengths, and annotation guidelines. We use them to evaluate the off-the-shelf performance of models trained on a single dataset. While they perform well within-domain (e.g., a new state-of-the-art of 79.3 F1 on LitBank), they still perform poorly out-of-domain.

To address poor out-of-domain performance, we propose joint training for coreference resolution, which is challenging due to the incompatible training procedues for different datasets. Among other things, we need to address (unannotated) singleton clusters, as OntoNotes does not include singleton annotations. We propose a data augmentation process to add predicted singletons, or *pseudo-singletons*, into the training data to match the other datasets which have gold singleton annotations.

Concretely, we contribute a benchmark for coreference to highlight the disparity in model performance and track generalization. We find joint training highly effective and show that including more datasets is almost "free", as performance on any single dataset is only minimally affected by joint training. We find that our data augmentation method of adding pseudo-singletons is also effective. With all of these extensions, we increase the macro average

*Equal Contribution

[1] Code available at https://github.com/shtoshni92/fast-coref

Proceedings of the 4th Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2021), pages 111–120
Punta Cana, Dominican Republic, November 10–11, 2021. ©2021 Association for Computational Linguistics

Dataset	Num. Docs Train	 Dev.	 Test	Words/ Doc	Mentions/ Doc	Mention length	Cluster size	% of singleton mentions
OntoNotes	2802	343	348	467	56	2.3	4.4	0.0
LitBank^k	80	10	10	2105	291	2.0	3.7	19.8
PreCo	36120	500	500	337	105	2.7	1.6	52.0
Character Identification	987	122	192	262	36	1.0	5.1	6.4
WikiCoref	0	0	30	1996	230	2.6	5.0	0.0
Quiz Bowl Coreference^k	0	0	400	126	24	2.7	2.0	26.0
Gendered Ambiguous Pronouns ^{p}	2000	400	2000	95	3	2.0	-	-
Winograd Schema Challenge^p	0	0	271	16	3	1.5	-	-

Table 1: Statistics of datasets. Datasets with ^{k} indicate that prior work uses k-fold cross-validation; we record the splits used in this work. Datasets with ^{p} are partially annotated, so we do not include cluster details.

F_1 across all datasets by 9.5 points and achieve a new state-of-the-art on LitBank and WikiCoref.

2 Datasets

We organize our datasets into three types. Training datasets (Sec. 2.1) are large in terms of number of tokens and clusters and more suitable for training. Evaluation datasets (Sec. 2.2) are out-of-domain compared to our training sets and are entirely held out. Analysis datasets (Sec. 2.3) contain annotations aimed at probing specific phenomena. Table 1 lists the full statistics.

2.1 Training Datasets

OntoNotes 5.0 (ON) (Weischedel et al., 2013) is a collection of news-like, web, and religious texts spanning seven distinct genres. Some genres are transcripts (phone conversations and news). As the primary training and evaluation set for developing coreference resolution models, many features specific to this corpus are tightly integrated into publicly released models. For example, the metadata includes information on the document genre and the speaker of every token (for spoken transcripts). Notably, it does not contain singleton annotations.

LitBank (LB) (Bamman et al., 2020) is a set of public domain works of literature drawn from Project Gutenberg. On average, coreference in the first 2,000 tokens of each work is fully annotated for six entity types.[2] We only use the first cross-validation fold of LitBank, which we call LB_0.

PreCo (PC) (Chen et al., 2018) contains documents from reading comprehension examinations, each fully annotated for coreference resolution. Notably, the corpus is the largest such dataset released.

2.2 Evaluation Datasets

Character Identification (CI) (Zhou and Choi, 2018) has multiparty conversations derived from TV show transcripts. Each scene in an episode is considered a separate document. This character-centric dataset only annotates mentions of people.

WikiCoref (WC) (Ghaddar and Langlais, 2016) contains documents from English Wikipedia. This corpus contains sampled and annotated documents of different lengths, from 209 to 9,869 tokens.

Quiz Bowl Coreference (QBC) (Guha et al., 2015) contains questions from Quiz Bowl, a trivia competition. These paragraph-long questions are dense with entities. Only certain entity types (titles, authors, characters, and answers) are annotated.

2.3 Analysis Datasets

Gendered Ambiguous Pronouns (GAP) (Webster et al., 2018) is a corpus of ambiguous pronoun-name pairs derived from Wikipedia. While only pronoun-name pairs are annotated, they are provided alongside their full-document context. This corpus has been previously used to study gender bias in coreference resolution systems.

Winograd Schema Challenge (WSC) (Levesque et al., 2012) is a challenge dataset for measuring common sense in AI systems.[3] Unlike the other datasets, each document contains one or two sentences with a multiple-choice question. We manually align the multiple choices to the text and remove 2 of the 273 examples due to plurals.

[2] They are people, facilities, locations, geopolitical entities, organizations, and vehicles.

[3] https://cs.nyu.edu/~davise/papers/WinogradSchemas/WSCollection.html

Model	Training	ON	LB$_0$	PC	CI	WC	QBC	GAP	WSC	Macro Avg.
longdoc	ON	79.0	54.8	44.3	49.8	59.6	36.8	88.9	59.8	59.1
longdoc ^{S}	ON	79.6	54.6	44.0	58.7	60.1	36.4	**89.8**	59.4	60.3
longdoc $^{S,\,G}$	ON	79.5	54.7	44.5	59.5	59.9	37.0	89.0	58.7	60.3
longdoc ^{S}	ON + PS 60K	**80.6**	56.6	49.1	55.6	62.1	40.1	89.3	61.3	**61.8**
longdoc	LB$_0$	56.6	77.2	46.8	53.3	47.5	**50.5**	85.3	32.8	56.3
longdoc	PC	58.8	50.3	**87.8**	39.5	50.7	46.5	87.3	**62.7**	60.5
longdoc ^{S}	Joint	79.2	**78.2**	87.6	59.4	60.3	42.9	88.6	60.1	69.5
longdoc ^{S}	Joint + PS 30K	79.6	**78.2**	87.5	58.4	**62.5**	45.5	88.7	59.4	**70.0**

Table 2: Performance of each model on 8 datasets measured by CoNLL F$_1$ (Pradhan et al., 2012), except for GAP (F$_1$) and WSC (accuracy). Some models use speaker (^{S}) features, genre (^{G}) features, or pseudo-singletons (PS).

3 Models

3.1 Baselines

We first evaluate a recent system (Xu and Choi, 2020) which extends a mention-ranking model (Lee et al., 2018) by making modifications in the decoding step. We find disappointing out-of-domain performance and difficulties with longer documents present in LB$_0$ and WC (Appendix B.1). To overcome this issue, we study the *longdoc* model by Toshniwal et al. (2020), which is an entity-ranking model designed for long documents that reported strong results on both OntoNotes and LitBank.

The original longdoc model uses a pretrained SpanBERT (Joshi et al., 2020) encoder which we replace with Longformer-large (Beltagy et al., 2020) as it can incorporate longer context. We retrain the longdoc model and finetune the Longformer encoder for each dataset, which proves to be competitive for coreference.[4] For OntoNotes we train with and without the metadata of: (a) genre embedding, and (b) speaker identity which is introduced as part of the text as in Wu et al. (2020).

3.2 Joint Training

With copious amounts of text in OntoNotes, PreCo, and LitBank, we can train a joint model on the combined dataset. However, this is impractical as the annotation guidelines between the datasets are misaligned (OntoNotes does not annotate singletons and uses metadata) and because there are substantially more documents in PreCo.

Augmenting Singletons Since OntoNotes does not annotate for singletons, our training objective for OntoNotes is different from that of PreCo and LitBank. To address this, we introduce *pseudo-singletons* that are *silver* mentions derived from first training a mention detector on OntoNotes and selecting the top-scoring mentions outside the gold mentions.[5] We experiment with adding 30K, 60K, and 90K pseudo-singletons (in total, there are 156K gold mentions). We find adding 60K to be the best fit for OntoNotes-only training, and 30K is the best for joint training (Appendix B.3).

Data Imbalance PreCo has 36K training documents, compared to 2.8K and 80 training documents for OntoNotes and LitBank respectively. A naive dataset-agnostic sampling strategy would mostly sample PreCo documents. To address this issue, we downsample OntoNotes and PreCo to 1K documents per epoch. Downsampling to 0.5K documents per epoch led to slightly worse performance (Appendix B.4).

Metadata Embeddings For the joint model to be applicable to unknown domains, we avoid using any domain or dataset-identity embeddings, including the OntoNotes genre embedding. We do make use of speaker identity in the joint model because: (a) this is possible to obtain in conversational and dialog data, and (b) it does not affect other datasets that are known to be single-speaker at test time.

4 Results

Table 2 shows the results for all our models on all 8 datasets. We report each dataset's associated metric (e.g., CoNLL F$_1$) and a macro average across all eight datasets to compare generalizability.

Among the longdoc baseline models trained on one of OntoNotes, PreCo, or LitBank, we observe a sharp drop in out-of-domain evaluations. The LitBank model is generally substantially worse than the models trained on OntoNotes and PreCo, likely

[4] The model scores 79.5 on OntoNotes and achieves state-of-the-art on LitBank with 79.3. Details are in Appendix B.2.

[5] This mention detector is architecturally the first half of the longdoc model.

Dataset	Instance
(1) QBC	(**This poem**) is often considered the counterpart of another poem . . . name this poem about a creature "burning bright, in the forests of the night," . . .
(2) QBC	This author's non fiction works . . . another work , a plague strikes secluded valley where teenage boys have been evacuated . . . name this author of Nip the Buds, Shoot the Kids . . .
(3) QBC	This poet of "(**I**) felt a Funeral in (**my**) Brain" and "I'm Nobody, Who are you?" wrote about a speaker who hears a Blue, uncertain, stumbling buzz before expiring in "(**I**) heard a fly buzz when (**I**) died". For 10 points, name this female American poet of Because (**I**) could not stop for Death.
(4) CI	*Chandler Bing:* Okay, I don't sound like that. (**That**) is so not true. (**That**) is so not ... (**That**) is so not ... That ... Oh , shut up !

Table 3: Joint + PS 30K error analysis for zero-shot evaluation sets. Each row highlights one cluster where spans in parenthesis are predicted by the model while the blue-colored spans represent ground truth annotations. Thus, in (2) the model misses out on the ground truth cluster entirely while in (3) and (4) the model predicts an additional cluster.

due to both a smaller training set and a larger domain shift. Interestingly, the LitBank model performs the best among all models on QBC, which can be attributed to both LB and QBC being restricted to a similar set of markable entity types. Meanwhile, all OntoNotes-only models perform well on WC and GAP, possibly due to the more diverse set of genres within ON and because WC also does not contain singletons.

For models trained on OntoNotes, we find that the addition of speaker tokens leads to an almost 9 point increase on CI, which is a conversational dataset, but has little impact for non-conversational evaluations. Surprisingly, the addition of genre embeddings has almost no impact on the overall evaluation.[6] Finally, the addition of pseudo-singletons leads to consistent significant gains across almost all the evaluations, including OntoNotes.

The joint models, which are trained on a combination of OntoNotes, LitBank, and PreCo, suffer only a small drop in performance on OntoNotes and PreCo, and achieve the best performance for LitBank. Like the results observed when training with only OntoNotes, we see a significant performance gain with pseudo-singletons in joint training as well, which justifies our intuition that they can bridge the annotation gap. The "Joint + PS 30K" model also achieves the state of the art for WC.

5 Analysis

Impact of Singletons Singletons are known to artificially boost the coreference metrics (Kübler and Zhekova, 2011), and their utility for downstream applications is arguable. To determine the impact of singletons on final scores, we present separate results for singleton and non-singleton clusters in QBC in Table 4. For non-singleton clusters we use the standard CoNLL F_1 but for singleton clusters the CoNLL score is undefined, and hence, we use the vanilla F_1-score.

The poor performance of ON models for singletons is expected, as singletons are not seen during training. Adding pseudo-singletons improves the performance of both the ON and the Joint model for singletons. Interestingly, adding pseudo-singletons also leads to a small improvement for non-singleton clusters.

The PC model has the best performance for non-singleton clusters while the LB_0 model, which performs the best in the overall evaluation, has the worst performance for non-singleton clusters. This means that the gains for the LB_0 model can be all but attributed to the superior mention detection performance which can be explained by the fact that both LB and QBC are restricted to a similar set of markable entity types.

Impact of Domain Shift Table 3 presents instances where the Joint + PS 30K model makes mistakes. In examples (1) and (2), the model misses out on mentions referring to literary works which

[6] In fact, we find that for the model trained with genre embeddings, modifying the genre value during inference has almost no impact on the final performance.

Data	Singleton	Non-singleton	Overall
ON	0.4	43.9	36.4
ON + PS 60K	14.4	44.4	40.1
LB_0	**44.9**	41.2	**50.5**
PC	28.8	**50.3**	46.5
Joint	21.7	47.3	42.9
Joint + PS 30K	26.7	48.6	45.5

Table 4: Performance on singleton and non-singleton clusters for QBC. ON=$\text{longdoc}^{\text{S}}$ and PS=pseudo-singletons.

is because references to literary texts are rare in the joint training data. Example (2) also requires world knowledge to make a connection between the description of the work and its title. In example (3) the model introduces an extraneous cluster consisting of first person pronouns mentioned in titles of different works. The model lacks the domain knowledge that narrators across different works are not necessarily related. Apart from the language shift, there are annotation differences across datasets as well. For example (4) drawn from CI, the model predicts a valid cluster (for Chandler Bing's speaking style) according to the ON annotation guidelines but the CI dataset doesn't annotate such clusters.

6 Related work

Joint training is commonly used in NLP for training robust models, usually aided by learning dataset, language, or domain embeddings (e.g., (Stymne et al., 2018) for parsing; (Kobus et al., 2017; Tan et al., 2019) for machine translation). This is essentially what models for OntoNotes already do with genre embeddings (Lee et al., 2017). Unlike prior work, our test domains are unseen, so we cannot learn test-domain embeddings.

For coreference resolution, Aralikatte et al. (2019) augment annotations using relation extraction systems to better incorporate world knowledge, a step towards generalization. Subramanian and Roth (2019) use adversarial training to target names, with improvements on GAP. Moosavi and Strube (2018) incorporate linguistic features to improve generalization to WC. Recently, Zhu et al. (2021) proposed the OntoGUM dataset which consists of multiple genres. However, compared to the datasets used in our work, OntoGUM is much smaller, and is also restricted to a single annotation scheme. To the best of our knowledge, our work is the first to evaluate generalization at scale.

Missing singletons in OntoNotes has been previously addressed through new data annotations, leading to the creation of the ARRAU (Poesio et al., 2018) and PreCo (Chen et al., 2018) corpora. While we include PreCo in this work, ARRAU contains additional challenges, like split-antecedents, that further increase the heterogeneity, and its domain overlaps with OntoNotes. Pipeline models for coreference resolution that first detect mentions naturally leave behind unclustered mentions as singletons, although understanding singletons can also improve performance (Recasens et al., 2013).

Recent end-to-end neural models have been evaluated on OntoNotes, and therefore conflate "not a mention" with "is a singleton" (Lee et al., 2017, 2018; Kantor and Globerson, 2019; Wu et al., 2020). For datasets with singletons, this has been addressed explicitly through a cluster-based model (Toshniwal et al., 2020; Yu et al., 2020). For those without, they can be implicitly accounted for with auxiliary objectives (Zhang et al., 2018; Swayamdipta et al., 2018). We go one step further by augmenting with *pseudo-singletons*, so that the training objective is identical regardless of whether the training set contains annotated singletons.

7 Conclusion

Our eight-dataset benchmark highlights disparities in coreference resolution model performance and tracks cross-domain generalization. Our work begins to address cross-domain gaps, first by handling differences in singleton annotation via data augmentation with pseudo-singletons, and second by training a single model jointly on multiple datasets. This approach produces promising improvements in generalization, as well as new state-of-the-art results on multiple datasets. We hope that future work will continue to use this benchmark to measure progress towards truly general-purpose coreference resolution.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Award No. 1941178.

References

Berfin Aktaş, Veronika Solopova, Annalena Kohnert, and Manfred Stede. 2020. Adapting Coreference Resolution to Twitter Conversations. In *Findings of EMNLP*.

Rahul Aralikatte, Heather Lent, Ana Valeria Gonzalez, Daniel Herschcovich, Chen Qiu, Anders Sandholm, Michael Ringaard, and Anders Søgaard. 2019. Rewarding Coreference Resolvers for Being Consistent with World Knowledge. In *EMNLP-IJCNLP*.

David Bamman, Olivia Lewke, and Anya Mansoor. 2020. An Annotated Dataset of Coreference in English Literature. In *LREC*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv:2004.05150*.

Hong Chen, Zhenhua Fan, Hao Lu, Alan Yuille, and Shu Rong. 2018. PreCo: A Large-scale Dataset in Preschool Vocabulary for Coreference Resolution. In *EMNLP*.

Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2018. Neural Models for Reasoning over Multiple Mentions Using Coreference. In *NAACL-HLT*.

Yifan Gao, Piji Li, Irwin King, and Michael R. Lyu. 2019. Interconnected Question Generation with Coreference Alignment and Conversation Flow Modeling. In *ACL*.

Abbas Ghaddar and Phillippe Langlais. 2016. WikiCoref: An English Coreference-annotated Corpus of Wikipedia Articles. In *LREC*.

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 *NAACL-HLT*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *TACL*, 8.

Ben Kantor and Amir Globerson. 2019. Coreference Resolution with Entity Equalization. In *ACL*.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain Control for Neural Machine Translation. In *RANLP*.

Sandra Kübler and Desislava Zhekova. 2011. Singletons and Coreference Resolution Evaluation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end Neural Coreference Resolution. In *EMNLP*.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-Order Coreference Resolution with Coarse-to-Fine Inference. In *NAACL-HLT*.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd Schema Challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*.

Manling Li, Alireza Zareian, Ying Lin, Xiaoman Pan, Spencer Whitehead, Brian Chen, Bo Wu, Heng Ji, Shih-Fu Chang, Clare Voss, Daniel Napierski, and Marjorie Freedman. 2020. GAIA: A Fine-grained Multimedia Knowledge Extraction System. In *ACL: System Demonstrations*.

Nafise Sadat Moosavi and Michael Strube. 2018. Using Linguistic Features to Improve the Generalization Capability of Neural Coreference Resolvers. In *EMNLP*.

Massimo Poesio, Yulia Grishina, Varada Kolhatkar, Nafise Moosavi, Ina Roesiger, Adam Roussel, Fabian Simonjetz, Alexandra Uma, Olga Uryupina, Juntao Yu, and Heike Zinsmeister. 2018. Anaphora Resolution with the ARRAU Corpus. In *Proceedings of the First Workshop on Computational Models of Reference, Anaphora and Coreference*.

Corbèn Poot and Andreas van Cranenburgh. 2020. A Benchmark of Rule-Based and Neural Coreference Resolution in Dutch Novels and News. In *CRAC*.

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*.

Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. 2013. The Life and Death of Discourse Entities: Identifying Singleton Mentions. In *NAACL-HLT*.

Sara Stymne, Miryam de Lhoneux, Aaron Smith, and Joakim Nivre. 2018. Parser Training with Heterogeneous Treebanks. In *ACL*.

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*.

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. 2018. Syntactic Scaffolds for Semantic Structures. In *EMNLP*.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual Neural Machine Translation with Language Clustering. In *EMNLP-IJCNLP*.

Shubham Toshniwal, Sam Wiseman, Allyson Ettinger, Karen Livescu, and Kevin Gimpel. 2020. Learning to Ignore: Long Document Coreference with Bounded Memory Neural Networks. In *EMNLP*.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the GAP: A Balanced Corpus of Gendered Ambiguous Pronouns. *TACL*, 6.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. OntoNotes Release 5.0. *Linguistic Data Consortium*.

Sam Wiseman, Alexander M. Rush, and Stuart Shieber. 2016. Antecedent Prediction Without a Pipeline. In *Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON)*.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. CorefQA: Coreference Resolution as Query-based Span Prediction. In *ACL*.

Patrick Xia, João Sedoc, and Benjamin Van Durme. 2020. Incremental Neural Coreference Resolution in Constant Memory. In *EMNLP*.

Liyan Xu and Jinho D. Choi. 2020. Revealing the Myth of Higher-Order Inference in Coreference Resolution. In *EMNLP*.

Jian Bo Yang, Qi Mao, Qiao Liang Xiang, Ivor Wai-Hung Tsang, Kian Ming Adam Chai, and Hai Leong Chieu. 2012. Domain Adaptation for Coreference Resolution: An Adaptive Ensemble Approach. In *EMNLP*.

Juntao Yu, Alexandra Uma, and Massimo Poesio. 2020. A Cluster Ranking Model for Full Anaphora Resolution. In *LREC*.

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 *ACL*.

Shanheng Zhao and Hwee Tou Ng. 2014. Domain Adaptation with Active Learning for Coreference Resolution. In *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)*.

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*.

Yilun Zhu, Sameer Pradhan, and Amir Zeldes. 2021. OntoGUM: Evaluating Contextualized SOTA Coreference Resolution on 12 More Genres. In *ACL*.

A Model and Training Details

A.1 Model

Our model follows the typical coreference pipeline of encoding the document, followed by mention proposal, and finally mention clustering. The model is architecturally the same as Toshniwal et al. (2020), and so we re-present their formulation throughout this section. However, we use the Longformer encoder as it accommodates longer documents. Otherwise, the model is identical to Toshniwal et al. (2020) in terms of model size and weight dimensions. We next explain the mention proposal and mention clustering stages briefly.

Mention Proposal Given a document $\mathcal{D}$, we score all mentions of length $\leq$ 20 subword tokens and choose the $K = 0.4 \times |\mathcal{D}|$ top spans among them. This is an initial pruning step that speeds up the model and reduces memory usage. Let $X(K) = \{(x_i)_{i=1}^{K}\}$ represent the top-K candidate mention spans and $s_m(x_i)$ be a learned scoring function for span x_i, which represents how likely a span is an entity mention. s_m is trained to assign positive score to gold mentions (any mention in a gold cluster), and negative score otherwise. The training objective only uses spans in $X(K)$, i.e. loss is computed after pruning. During inference, we can therefore further prune down to $\{x_i : x_i \in X(K), s_m(x_i) \geq 0\}$, which we then pass into the clustering step. During training, we use teacher forcing and only pass gold mentions among the top-K mentions to the clustering step.

Mention Clustering The entity-based model tracks M entities (initially $M = 0$). Let $E = (e_m)_{m=1}^{M}$ represent the M entities. For ease of notation, we will overload the terms x_i and e_j to also correspond to their respective representations.

The model decides whether the span x_i refers to any of the entities in E as follows:

$$\begin{aligned} s_c(x_i, e_j) &= f_c([x_i; e_j; x_i \odot e_j; g(x_i, e_j)]) \\ s_c^{top} &= \max_{j=1 \ldots M} s_c(x_i, e_j) \\ e^{top} &= \arg\max_{j=1 \ldots M} s_c(x_i, e_j) \end{aligned}$$

where $\odot$ represents the element-wise product, and $f_c(\cdot)$ corresponds to a learned feedforward neural network. The term $g(x_i, e_j)$ corresponds to a concatenation of feature embeddings that includes embeddings for (a) number of mentions in e_j, and (b) number of tokens between x_i and last mention of e_j. If $s_c^{top} > 0$ then x_i is considered to refer to e^{top}, and e^{top} is updated accordingly.[7] Otherwise, we initiate a new cluster: $E = E \cup x_i$. During training, we use teacher-forcing i.e. the clustering decisions are based on ground truth.

A.2 Training

We train all the models for 100K gradient steps with a batch size of 1 document. Only the LB-only models are trained for 8K gradient steps which corresponds to 100 epochs for LB. The models are evaluated a total of 20 times (every 5K training steps) for all models except the LB-only models which are evaluated every 400 steps. We use early stopping and a patience of 5 i.e. training stops if the validation performance doesn't improve for 5 consecutive evaluations.

We use the full context size of 4096 tokens for Longformer-large. All training documents used in this work except 1 ON document fit in a single context window. For optimizer, we use AdamW with a weight decay of 0.01 and initial learning rate of 1e-5 for the Longformer encoder, and Adam with an initial rate of 3e-4 for the rest of the model parameters. The learning rate is linearly decayed throughout the training.

B Other Results

B.1 Xu and Choi (2020) Baselines

We run the off-the-shelf model on the test sets of ON, LB$_0$, PC, and QBC. LB$_0$ requires a 24GB GPU, while WC runs out of memory even on that hardware. The model shows strong in-domain performance with 80.2 on ON. However, out-of-domain performance is weak: 57.2 on LB$_0$, 49.3 on PC, and 37.6 on QBC. These are roughly on par with the ON longdoc models.

B.2 LitBank Cross-Validation Results

Table 7 presents the results for all the cross-validation splits of LitBank. The overall performance of 79.3 CoNLL F1 is state of the art for LitBank, outperforming the previous state of the art of 76.5 by Toshniwal et al. (2020). Note that in this work, the joint model outperformed (78.2 vs. 77.2) this baseline model on split 0 (LB$_0$). However, training 10 joint models contradicts the purpose of this work, which is to create a single, generaliz-

[7] We use weighted averaging where the weight for e^{top} corresponds to the number of previous mentions seen for e^{top}.

Training			Evaluation				
Num ON	Num PC	Num PS	ON	LB	PC	WC	QB
500	500	0	79.4	78.8	85.0	60.8	45.3
500	500	30K	79.4	79.5	84.8	61.2	47.7
1000	1000	0	79.7	79.4	85.1	60.3	42.9
1000	1000	30K	79.5	78.7	85.1	62.5	45.5
1000	1000	60K	78.9	77.4	85.1	61.3	46.6
1000	1000	90K	78.5	77.7	85.1	60.7	47.4

Table 5: Validation set scores of datasets when downsampling OntoNotes (ON) and PreCo (PC) in joint training.

	Training	ON	LB	PC	CI	WC	QBC	Macro Avg.
longdoc	ON	88.8	62.8	42.6	65.7	72.1	53.1	64.2
longdoc ^{S}	ON	89.2	61.7	41.8	75.5	69.9	52.5	65.1
longdoc $^{S, G}$	ON	89.4	62.8	42.7	77.0	72.5	52.9	66.2
longdoc ^{S}	ON + PS 60K	87.7	81.5	81.0	76.1	71.6	70.4	78.0
longdoc	LB_0	77.6	85.8	81.4	67.6	69.8	70.2	75.4
longdoc	PC	76.8	81.1	93.6	66.5	67.0	77.3	77.0
longdoc ^{S}	Joint	90.9	86.9	93.4	77.2	76.6	68.7	82.3
longdoc ^{S}	Joint + PS 30K	89.4	87.1	93.3	77.3	74.6	71.7	82.2

Table 6: Result of all the models with gold mentions. Some models use speaker (^{s}) features, genre (^{g}) features, or pseudo-singletons (PS). The metric for the training and evaluation datasets is CoNLL F-score. We skip the analysis datasets because they lack the set of true gold mentions.

Cross-val split	Dev	Test
0	78.8	77.2
1	78.6	80.3
2	81.1	78.7
3	79.0	79.1
4	80.0	78.7
5	79.7	78.7
6	77.7	80.7
7	81.9	79.1
8	78.0	80.8
9	81.8	78.7
Total	79.7	79.3

Table 7: LitBank cross-validation results.

Num PS	Val.	Test
0	79.9	79.6
30K	79.9	80.5
60K	**80.0**	**80.6**
90K	79.9	80.0

Table 8: Validation and Test results for ON-only model trained on different amount of pseduo-singletons (PS).

able model. Realistically, we recommend jointly training with the entirely of LitBank.

B.3 Singleton Results for OntoNotes

For ON-only models, we tune over the number of pseudo-singletons sampled among {30K, 60K, 90K}. Table 8 shows that 60K pseudo-singletons is the best choice based on validation set results on ON.

B.4 Downsampling and Singleton Results for Joint

In preliminary experiments, we sample 500 docs from ON and PC. Table 5 shows the results, confirming that 1K is slightly better than 500. Using more examples (e.g. 5K PC) begins to hurt performance on LB, likely due to data imbalance.

For the 1K downsampling setting, we tune over the number of pseudo-singletons sampled among {30K, 60K, 90K}. We find 30K to be the best choice based on validation set results.

B.5 Results with Gold Mentions

In Table 6, we report the results with gold mentions for the training and evaluation sets. The analysis sets are skipped as they are partially annotated. We find that joint training is also helpful in this setting, as results mirror findings with predicted mentions. In particular, this shows that it is not just a failure to predict mentions that is preventing ON from performing well on LB, PC, and QBC.

C Compute Resources

Given that we are finetuning the Longformer model and using a maximum context size of 4096 tokens, the memory requirements of the model are quite large even though the cluster-ranking paradigm is considered memory efficient (Xia et al., 2020). We were able to train the PreCo-only model on a 12 GB GPU in 20 hrs (even the longest PreCo documents are shorter than 2048 tokens with the Longformer tokenization). All other models were trained over GPUs with memory 24GB or higher (Titan RTX and A6000). On an A6000, the LB-only models can be trained within 4 hrs, the ON-only models within 16 hrs, and the joint models within 20 hrs.