Coreference in Long Documents using Hierarchical Entity Merging

Talika Gupta[†*] and **Hans Ole Hatzel** [‡] and **Chris Biemann** [‡]
[†] IIIT Guwahati, Assam, India
[‡] Language Technology Group, Universität Hamburg, Germany
talika.gupta@iiitg.ac.in,
{hans.ole.hatzel, chris.biemann}@uni-hamburg.de

Abstract

Current top-performing coreference resolution approaches are limited with regard to the maximum length of texts they can accept. We explore a recursive merging technique of entities that allows us to apply coreference models to texts of arbitrary length, as found in many narrative genres. In experiments on established datasets, we quantify the drop in resolution quality caused by this approach. Finally, we use an under-explored resource in the form of a fully coreference-annotated novel to illustrate our model's performance for long documents in practice. Here, we achieve state-of-the-art performance, outperforming previous systems capable of handling long documents.

1 Introduction

Coreference resolution has significant time and memory requirements which, in currently released models, typically increase at least quadratically with the length of the document, resulting in inefficient systems. These substantial computational requirements make coreference resolution impractical for long documents such as novels. The task of establishing coreference links in such texts is important to enable a wide range of downstream tasks such as extracting character interaction networks (e.g. Konle and Jannidis, 2022). We propose a novel hierarchical algorithm for coreference resolution, to conserve computational resources while still achieving good performance. Our approach allows the models to – in principle – scale to documents of arbitrary length, outperforming existing long-document approaches.

Our proposed approach works by splitting a long document into multiple splits and then running an existing coreference resolution model on each split, thereby extracting the entities in each of them. We then propose a merging approach, where we pairwise merge the entities across splits by leveraging existing mention linking models, applying them to the merging of clusters instead. We experiment by splitting a document into a varying number of constituent parts and document the effectiveness of the merging approach as the length of the split decreases.

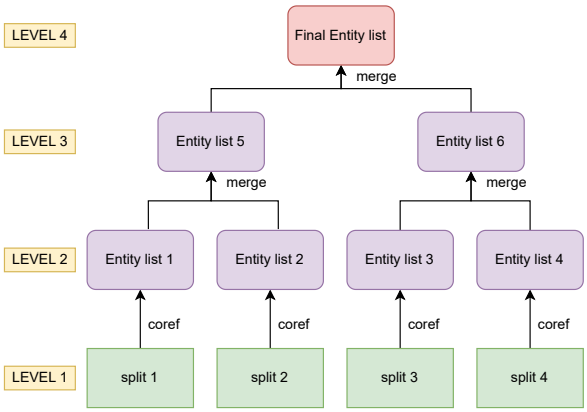

Figure 1: Our hierarchical merging approach iteratively merges pairs of entity lists until we arrive at a single set of entities that spans the full document. The first sets of weight denoted by "coref" is used while generating entity lists in each split. The second sets of weight denoted by "merge" is used while merging the entity lists across splits.

2 Background & Related Work

Coreference resolution is the task of identifying corefering spans in a text, that is to say, those groups of spans that refer to the same entity. Traditional coreference resolution comprises of two phases: span extraction from the text and the subsequent identification of coreference links among the extracted spans. The Word-Level Coreference (subsequently wl-coref) Resolution model (Dobrovolskii, 2021) separates the task of coreference resolution from span extraction and solves it on the word level, hence lowering the time complexity of the

[*]Work conducted as part of an internship at Universität Hamburg

model to $O(n^2)$, where n is the length of the document. The span extraction is performed separately only for those words that are found to be coreferent to some other words. We will base our experiments on this model, but note that our hierarchical approach can in principle also be used with the other major coreference resolution model architectures, with the only requirement being that mentions can be represented by fixed-length embeddings. The models by Bohnet et al. (2023); Zhang et al. (2023) are recent sequence-to-sequence coreference models, but are impractical for long documents due to their memory requirements, with even short document processing being very resource-intensive. Our proposed hierarchical merging strategy could potentially help to apply these models to long documents.

We train our English model on the OntoNotes dataset (Pradhan et al., 2012) and the LitBank (Bamman et al., 2020) dataset, the latter containing coreference annotations for literary texts. We evaluate our approach on LitBank, and observe competitive performance to the state-of-the-art coreference models while being memory efficient. For training our German models, we employ the TüBa-DZ news dataset (Telljohann et al., 2004) and the DROC literature dataset (Krug et al., 2018).

Efficient coreference resolution in long documents is a task that current models struggle with, due to their memory-intensive nature. Traditional incremental coreference resolution models use global entity representations, but their performance lags behind compared to other models (Toshniwal et al., 2020). Thirukovalluru et al. (2021) propose a scalable coreference resolution approach that works on the token level instead of the span level, and drops non-essential candidate antecedents to improve memory and time requirements. They test their system on a long book for which no annotations are available. Their code is not available as open source, which prevented us from testing it on the full book data that we use.

3 Methodology

In this section, we describe our coreference approach in detail. We build upon the wl-coref model (Dobrovolskii, 2021), picking it over alternatives due to its quick inference and relative simplicity in conjunction with competitive performance. Our code is publicly available.[*] We split each document evenly into n splits, ensuring an equal number of sentences in each, where $n \in \{2, 4, 8\}$. Treating each split as an independent document, the model performs inference as normal on each of them, ideally yielding one cluster for each entity in each split. That is to say, the model predicts pairwise scores for whether two words co-refer and builds a transitive closure over those connections that exceed a certain threshold.

Now, since some entities in one split may refer to the same entity as entities in another split, we develop a merging approach to merge such coreferring entities across splits. This process is similar to linking two mentions, except that instead of passing mention embeddings, we pass entity embeddings to the model. Hence, we need a way to represent individual entities as vectors. In this work, we only evaluate the approach of creating entity embeddings by means of averaging over all embeddings in a cluster.

Next, we merge the entities from two splits at a time, by passing the entity embeddings in the same way that the model takes in the word embeddings to create links among them. This results in an entity list that spans both the splits. The merging process is repeated for all such disjoint pairs of splits in each level, resulting in $n/2$ splits in the subsequent level if the previous level had n splits. Consequently, the last level of our hierarchical approach results in entities that encompass the entire document. This is illustrated in Figure 1.

4 Experiments

We conduct a set of experiments to evaluate the efficacy of the proposed hierarchical merging approach. First, we train the standard wl-coref model, which is pre-trained on the OntoNotes dataset, for an additional 10 epochs on the LitBank dataset. The resulting model is evaluated on the test split of LitBank, from which we exclude the singleton mentions since the original wl-coref architecture removes singletons during the span extraction step. That is to say we evaluate on a version of the texts without singleton mentions.

In terms of merging approaches, we follow the process laid out in Section 3. We document the results under three scenarios: **(a)** without merging entities across splits, **(b)** merging entities without

[*] https://github.com/uhh-lt/hierarchical-coref

training the model for merging (i.e. using the same model twice), and **(c)** merging entities after training the model for merging.

We expect the merging approach to have a negative impact on prediction quality, at least for short documents. To quantify said impact of merging on the prediction quality, we split the documents into n equal-sized splits and experiment with different values of n, specifically 2, 4, and 8.

We set a baseline result by refraining from merging the entities across splits. For the merging module, we experimented with and without training the model specifically for merging. For training data, we use 2-way splits and use gold entities in the individual splits, rather than predicted ones. We subsequently average the span embeddings in each entity to obtain the entity embeddings in each split. As these embeddings are handled analogously to word embeddings in the original setup, the model creates links between the entities in the same way that it does so for the words. We evaluate the entities now spanning the entire documents on the gold data. Our merging model is trained for 10 epochs using 2-way split documents, on top of the existing word-level weights.

As we found the model to lose its span prediction capabilities after training the merging module, we use two distinct sets of weights for the two tasks. For the first level of our approach, where the model generates an entity list, we employ the first sets of weights, which was trained on LitBank but not trained specifically for merging entities (this is denoted as "coref" in Figure 1). For subsequent levels of our approach, where entity lists from two splits are merged to produce an entity list that spans both the splits, we use the second set of weights, which was specifically trained for merging entities (indicated by "merge" in Figure 1). We refer to the recursive application of the merging step as hierarchical entity merging.

Our approach is primarily evaluated on the standard CoNLL-F1 score. Additionally, we provide LEA, an evaluation metric that focuses on coreference links and resolves several issues that CoNLL-F1 constituent scores suffer from (Moosavi and Strube, 2016).

4.1 German Data

The main advantage of our model lies in its modest memory consumption, enabling the processing of documents of arbitrary length. Accordingly,

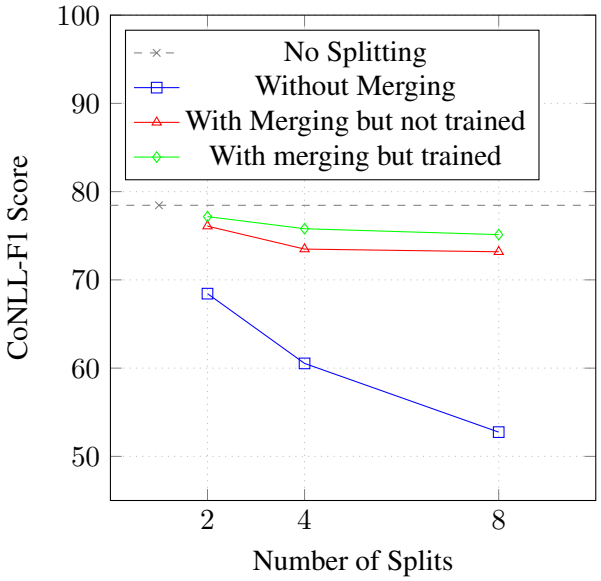

Figure 2: Comparison of CoNLL-F1 scores across varying numbers of splits, with and without merging the entity lists across splits

it is important to understand the performance on book-length literary texts. We are not aware of any such dataset in English, so instead we evaluate our model on two German texts: **(a)** the fairy tale "Der blonde Eckbert" by Ludwig Tieck (subsequently *Eckbert*) and **(b)** the full novel "Effi Briest" by Theodor Fontane (subsequently *Briest*). The two texts have around 7,000 and more than 100,000 tokens, respectively, and provide a good, if small-scale, benchmark for long-document coreference systems. For both novels, despite the coreference data being publicly available[*] we only know of results from a single system (Krug, 2020) on the data. For comparison, we provide the performance numbers of the system by Schröder et al. (2021) in Table 1, which was trained on German data, including the DROC corpus. Other models are generally not applicable to the full novel due to memory requirements. We train our hierarchical model on 2 and 4 splits of DROC, after training it on TüBa-D/Z using the German model gelectra-large as a foundation model (Chan et al., 2020). We chose 32 as a split size as it was the smallest number of splits to not cause memory errors at inference time for both texts. For comparison, we also provide numbers for 64 splits. Operating on split numbers that are not powers of two would also be possible

[*] https://gitlab2.informatik.uni-wuerzburg.de/kallimachos/AnnotierteDaten/-/tree/master/komplett

Text	System	CoNLL-F1	LEA
Eckbert	wl-coref	66.91	63.06
Eckbert	Schröder et al. (2021)	66.74	59.27
Briest	Schröder et al. (2021)	44.71	15.92
Briest	Krug (2020)	51.76	-

Table 1: Performance comparison of existing systems on the full stories. The system by Schröder et al. (2021) is the incremental system capable of handling arbitrary-length texts in constant memory. The rule-based system by Krug (2020) includes singletons in its evaluation and is therefore not directly comparable. Wl-coref is the word-level coreference model trained on DROC and TüBa-D/Z.

Text	Num-Splits	CoNLL-F1	LEA
Eckbert	32	51.56	59.06
Eckbert	64	57.27	50.78
Briest	32	52.89	36.75
Briest	64	54.17	39.12

Table 2: Performance of our system on the full German stories using either 32 or 64 splits of the original text. For detailed results see Appendix A, Table 5.

and would result in leaf nodes at varying depths of the merging tree. Our implementation omits this option for simplicity.

5 Results

All the scores have been obtained using the official CoNLL-2012 scorer (Pradhan et al., 2014). We perform experiments with varying numbers of splits, specifically splitting the document into 1 (i.e. not splitting it), 2, 4, and 8 segments, as can be seen in Figure 2.

For the unsplit document, we achieve a CoNLL-F1 score of 78.45. For two splits, we achieve a score of 76.06 if we employ merging without specifically training the merging module (using only first sets of weights). We achieve an improved F1 score of 77.30 if we additionally train the merging module (using two sets of weights). As a control experiment where we do not merge entity lists at all, we unsurprisingly obtain a much worse F1 score of 70.27. Similarly, for four and eight splits, the scores increase by two points on average as we go from one set of weights approach to the two sets of weights approach. The best F1 score we achieve is 74.52 and 73.96 respectively, by using two sets of weights. The scores when not conducting a merging step are unsurprisingly far below those with a merging step and also drastically decrease as we split the text into more sections.

Num Splits	Merger trained	CoNLL-F1	LEA
1	Not needed	78.45	**78.37**
2	appended	70.27	64.78
	✗	76.06	74.68
	✓	77.30	**76.17**
4	appended	60.54	50.20
	✗	73.49	71.53
	✓	75.80	**74.52**
8	appended	52.75	40.35
	✗	73.18	72.52
	✓	75.13	**73.96**

Table 3: LEA and CoNLL-F1 scores for a varying number of document splits on the LitBank data. For detailed results see Appendix A, Table 4.

Our results for "Der Blonde Eckbert" using the word-level system also illustrate that the incremental system yields competitive results for short texts. With regard to the full-book data "Effi Briest", our approach was able to set a new state-of-the-art result, outperforming the incremental model as well as the rule-based model by Krug (2020), in terms of both CoNLL-F1 and LEA scores. While the CoNLL-F1 score is already considerably better than with the incremental system, moving from 44.71 to 54.17 (see Table 2), we see a much more substantial improvement in the LEA score, from 15.92 to 39.12, which we attribute to LEA's approach of marking down split entities. Our model outperforms the rule-based model, moving from a 51.76 CoNLL-F1 score to 54.17, although results are not directly comparable because Krug (2020) includes singleton mentions in his evaluation. These results were attained by further training the merging module of our model, first using the four splits of "Effi Briest" and then the two splits. In this experiment, shorter splits, generally, have a positive impact on performance. We attribute this to our average-pooling approach, which presumably results in worse embeddings in longer texts.

In practice, our model is very run-time efficient and inference for the whole book "Effi Briest" takes just under two minutes on a single A6000 GPU.

6 Conclusion

Our results demonstrate a potential path for applying state-of-the-art models to longer text without vast increases in memory requirements. We use existing systems to provide a baseline result for

coreference resolution on a fully-annotated novel and set new state-of-the-art results. There is further headroom, as we are – for reasons of simplicity – not using the best available model for within-split resolution. Our training process could also be improved, for example by randomly splitting documents (rather than into roughly equal portions) to create more training data. While our approach exhibits reasonable coreference resolution quality for novels, it still does not suffice for use as a preprocessing step in most literary computing applications. We suspect that a substantial improvement of our approach would be possible by creating specialized embeddings for merging rather than averaging mention embeddings; additional error analysis will be needed to verify this intuition. Our general idea of hierarchical merging could also potentially be adapted to the very recent seq2seq coreference architectures (Zhang et al., 2023; Bohnet et al., 2023) where they could help overcome the model's maximum input lengths.

Acknowledgements

This work was, in part, funded through the project "Evaluating Events in Narrative Theory (EvENT)" (grant `BI 1544/11-1`) funded by the DFG.

References

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

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.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Leonard Konle and Fotis Jannidis. 2022. Modeling Plots of Narrative Texts as Temporal Graphs. In *Proceedings of the Computational Humanities Research Conference 2022*, volume 3290 of *CEUR Workshop Proceedings*, pages 318–336, Antwerp, Belgium. CEUR.

Markus Krug. 2020. *Techniques for the Automatic Extraction of Character Networks in German Historic Novels*. Ph.D. thesis, Julius Maximilians University Würzburg, Germany.

Markus Krug, Lukas Weimer, Isabella Reger, Luisa Macharowsky, Stephan Feldhaus, Frank Puppe, and Fotis Jannidis. 2018. Description of a corpus of character references in German novels-DROC [Deutsches ROman Corpus]. *DARIAH-DE Working Papers*, 27.

Nafise Sadat Moosavi and Michael Strube. 2016. Which Coreference Evaluation Metric Do You Trust? A Proposal for a Link-based Entity Aware Metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–642, Berlin, Germany. Association for Computational Linguistics.

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, USA. 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.

Fynn Schröder, Hans Ole Hatzel, and Chris Biemann. 2021. Neural end-to-end coreference resolution for German in different domains. In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021)*, pages 170–181, Düsseldorf, Germany. KONVENS 2021 Organizers.

Heike Telljohann, Erhard Hinrichs, and Sandra Kübler. 2004. The Tüba-D/Z Treebank: Annotating German with a Context-Free Backbone. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, pages 2229–2232, Lisbon, Portugal. European Language Resources Association.

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

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

Wenzheng Zhang, Sam Wiseman, and Karl Stratos. 2023. Seq2seq is all you need for coreference resolution. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11493–11504, Singapore. Association for Computational Linguistics.

A Full Results

		CoNLL				
Num Splits	Merger trained	**MUC**	**CEAFE**	**B^3**	**F1**	**LEA**
1	Not needed	90.06	65.80	79.55	78.45	78.37
2	appended	89.03	60.77	66.48	70.27	64.78
	✗	89.45	63.95	76.14	76.06	74.68
	✓	89.56	65.91	77.57	77.30	76.17
4	appended	87.12	52.26	52.56	60.54	50.20
	✗	88.63	60.65	73.14	73.49	71.53
	✓	88.94	63.74	75.98	75.80	74.52
8	appended	84.08	43.05	43.51	52.75	40.35
	✗	87.64	58.32	74.25	73.18	72.52
	✓	88.02	62.94	75.59	75.13	73.96

Table 4: Detailed results for our LitBank experiment.

		CoNLL				
Text	**System**	**MUC**	**CEAFE**	**B^3**	**F1**	**LEA**
Eckbert	wl-coref	93.17	28.23	67.79	66.91	63.06
Eckbert	Schröder et al. (2021)	93.80	46.44	59.97	66.74	59.27
Briest	Schröder et al. (2021)	86.79	29.19	18.16	44.71	15.92
Briest	Krug (2020)	85.8	29.9	39.6	51.76	-

Table 5: Detailed performance comparison of existing systems on the full German stories.

		CoNLL				
Text	**Num-Splits**	**MUC**	**CEAFE**	**B^3**	**F1**	**LEA**
Eckbert	32	91.67	32.80	52.71	51.56	59.06
Eckbert	64	91.74	28.38	51.69	57.27	50.78
Briest	32	89.96	31.22	37.50	52.89	36.75
Briest	64	90.24	32.40	39.87	54.17	39.12

Table 6: Performance of our system on the full document using varying number of initial splits.