Fine-Tuning Causal LLMs for Document-Level Literary Coreference Resolution

Anonymous ACL submission

Abstract

Coreference resolution is a key task in computational literary analysis. However, literary narratives present challenges for traditional coreference systems, most notably due to their extended document length. We fine-tuned decoder-only causal language models to perform coreference resolution on chunked literary texts. We then used the models as part of a custom pipeline that reconstructs the original document and resolves long-range coreference links. Our results show the effectiveness of both fine tuning causal LLMs for this task and our pipeline. Our approach outperforms current state-of-the-art methods across two literary coreference benchmark datasets.

1 Introduction

Coreference resolution is the task of determining when different mentions in a piece of text refer to the same entity. For example, consider the excerpt from George Eliot's Middlemarch shown in Figure 1. A successful coreference resolution system would recognise the mentions *Miss Brooke*, *her sister* and *Dorothea* all refer to the same entity, Dorothea Brooke, and the mentions *Celia* and *her* refer to the Celia Brooke. The set of coreferring mentions is described as being a coreference chain or cluster.

> ...but in **[Miss Brooke|0]**'s case, religion alone would have determined it; and **[Celia|1]** mildly acquiesced in all **[[her|1] sister|0]**'s sentiments, only infusing them with that common-sense which is able to accept momentous doctrines without any eccentric agitation. **[Dorothea|0]** knew many passages of...

Figure 1: Example of coreference entity annotation

In the field of digital humanities, coreference resolution is often a crucial first step in extracting information for further analysis. For example, to investigate relationships between characters (Nalisnick and Baird, 2013), explore gender representations (Underwood et al., 2018), or to provide narrative summaries (Gorinski and Lapata, 2015). Because many downstream tasks rely on accurate entity linking, the quality of the coreference system is of central importance.

Recently, it has been shown that generative large language models (LLMs) can achieve excellent performance in coreference resolution tasks (Zhang et al., 2023; Paolini et al., 2021). The architecture of LLMs make them well suited to dealing with the nuances and subtleties of language that effective coreference resolution requires.

Although significant progress has been made in domains such as news, for example using the OntoNotes dataset, the specific problem of coreference resolution in literary works (e.g. novels and screenplays) has received less attention. Literary texts pose particular problems that are not found in other written works. For example, their significant length and distance between coreferring mentions, frequently shifting perspectives, the use of generic and non-generic noun phrases, the evolution of characters, and the deliberate use of disguise and mistaken identity for comic or dramatic effect. See Rösiger et al. (2018) for a detailed breakdown.

This paper explores the use of decoder-only causal LLMs for literary coreference resolution. We identify two main challenges in using causal language models for this task. First, the well known hallucination problem, where the model erroneously inserts or omits tokens. Second, the difficulty in processing long documents, due to context-window limitations or the degradation of performance on extended input (Du et al., 2025). To address these issues, we have developed a coreference pipeline that implements token realignment, chunk merging, and document-level cluster reconciliation. We make the following contributions:

- The development of fine-tuned decoder-only large language models for coreference resolution.
- A practical long-document pipeline for aligning and merging chunked documents, and linking long-range coreference predictions [1].
- An evaluation of our approach across two benchmark literary coreference datasets LitBank (Bamman et al., 2019) and MovieCoref (Baruah and Narayanan, 2023).

2 Related Work

Initial approaches to identifying and resolving entities in the field of digital humanities e.g. Srivastava et al. (2016); Agarwal et al. (2013) implemented tasks such as character extraction using off-the-shelf tools such as Stanford Core NLP (Labatut and Bost, 2019). These works often focussed on the fundamental task of named entity recognition rather than the more challenging problem of unifying mentions. Consequently, key information, such as minor characters referred to only by noun phrases, was often absent from subsequent analysis (Vala et al., 2015).

In the wider field of coreference resolution, the use of neural models to evaluate text spans resulted in a significant step forward in accuracy (Lee et al., 2017). Initial results were further improved by researchers that used BERT for the initial embeddings (Joshi et al., 2019), encoded spans instead of individual words (Joshi et al., 2020), added further pretraining on top of BERT to enrich the embeddings with coreferential information (Ye et al., 2020), and by reframing coreference resolution as a span prediction task (Wu et al., 2020).

These early neural coreference systems had limited scalability. This led to further work aimed at reducing complexity requirements by avoiding expensive full span-based representations (Kirstain et al., 2021; Dobrovolskii, 2021). Further efficiencies focussed on restricting the number of antecedent comparisons (Toshniwal et al., 2020; Thirukovalluru et al., 2021); this allowed models to handle longer documents, and for the first time evaluations considered literary works. Toshniwal et al. (2020, 2021) evaluated models using the LitBank dataset. Baruah and Narayanan (2023) used Dobrovolskii's model to perform coreference resolution on full length screenplays.

In recent years focus has shifted away from span ranking to the use of generative language models to perform coreference resolution. Urbizu et al. (2020) used a Fairseq transformer model to generate coreference information for each input token. The approach was limited to small documents and a limited vocabulary. The results were significantly short of the existing systems. However, the research showed the potential of a text generation approach. Bohnet et al. (2023) used a T5 seq2seq model (Raffel et al., 2020). The input consisted of previous sentences annotated with coreference links along with the current sentence to be processed. The model outputs identified coreference links between the current sentence and previous sentences. They were able to achieve state-of-the-art accuracy on the CoNLL-2012 English dataset.

Researchers found that simpler approaches could also work effectively. Paolini et al. (2021) introduced the TANL framework, fine tuning a T5 model to replicate input text and augment it with annotations for a variety of NLP tasks, including coreference resolution. To handle longer documents, input documents were processed in chunks, after which a dynamic programming algorithm was used to reconstruct the document. They achieved competitive results on the CoNLL-12 dataset. However, there wasn't a strong focus on coreference resolution and there wasn't any consideration of literary works. Zhang et al. (2023) took a very similar approach. They used larger models, larger input chunks, and were able to achieve strong performance across a number of benchmark datasets, CoNLL-12 English and LitBank. There were limitations to their work; long documents such as complete screenplays or full-lengths novels weren't considered. Furthermore, few details were provided on how text chunks and their contained clusters were merged into complete documents. Finally, Hicke and Mimno (2024) focussed specifically on literary coreference resolution using LLMs. Again, they trained encoder-decoder models to replicate input and add annotations. However, they only considered single sentences, and there wasn't any attempt to link coreference links between different sentences.

Our work builds on these examples in two ways. First, instead of seq2seq models, we focus on decoder only, causal language models, for example Llama 3.1 (Team, 2024), Mistral Nemo (Mistral AI and NVIDIA, 2024), which represent a contemporary standard for task specific training. Second, we

[1] https://to-be-released-on-publication

consider the important issue of how to deal with complete, full length literary works; previous work on generative LLMs for coreference resolution only considers documents that are limited in length.

3 Method

We follow a similar approach Paolini et al. (2021), Zhang et al. (2023) and Hicke and Mimno (2024) in fine-tuning a generative language model to annotate model inputs with coreference annotations. Training examples consist of a plain text input sequence, and a target sequence that replicates the input text and inserts coreference markers that specify the boundaries of the mention and the cluster ID (see Figure 1 for an example). We use instruction tuned models. As such, we provide system prompts, along with the input sequence.

Unlike previous works that undertook full fine-tuning, we employ Parameter Efficient Fine Tuning (PEFT). By updating a small fraction of model weights; PEFT allows us to use larger base models and implement faster training cycles.

The trained model then forms part of our coreference resolution pipeline, which is shown in Figure 2. Full details on each stage of the piepline are provided in the following sections.

3.1 Chunking

The input document is split using natural structural boundaries. Novels are split into sentences. Screenplays are split on the boundaries between structural elements (speaker, dialogue and action descriptions). The intention here is to retain contextual information that may be otherwise lost, for example, if pronouns are separated from their antecedents. We then join these segments into initial chunks. Finally we apply a sliding-window approach that pairs adjacent chunks to create overlapping chunks.

3.2 Token Realignment

Even when fine-tuned, generative LLMs will not reliably replicate the input text. Downstream tasks such as evaluation require exact alignment with the original text. Therefore, following inference, we process the model output to realign it with the original text.

First, the model's output is parsed to extract predicted coreference entity spans. We then treat alignment as a token matching problem between tokens from the original text and tokens generated by the model. Tokens are iterated over in parallel; when a mismatch occurs, we search for the next shared fixed-length n-gram sequence (we used n=5) between the remaining target tokens and the remaining model tokens. The location of this match is used to calculate the extent of the insertion or deletion error. The coreference spans are then shifted or removed to restore alignment with the original text. After processing the entire sequence, the corrected sequence of tokens exactly matches the input document and the model generated tokens can be discarded. This simple approach avoids expensive dynamic programming that was used in previous similar work (Paolini et al. (2021); Zhang et al. (2023)) while remaining robust to hallucinations introduced by generative decoding.

3.3 Merging Chunks

Because documents are processed in overlapping chunks, the same coreference entity spans may appear in adjacent chunks but be assigned different local clusters IDs. Therefore, we propagate entity cluster IDs across overlapping regions prior to merging chunks.

Starting with the first pair of consecutive chunks, we identify entity spans present in the overlapping region and use the cluster IDs assigned in the earlier chunk and overwrite the IDs in the subsequent chunk. Following this propagation step, any remaining entity identifiers in the latter chunk are renumbered to avoid collisions and to preserve consecutive numbering. The chunks are then merged by retaining the non-overlapping portion of the later chunk. We repeat this process with subsequent chunks, eventually resulting in a single document with consistent entity identifiers. See Figure 2 for a concrete example.

3.4 Merging Clusters

Chunk-level merging recovers coreference links in adjacent chunks, but some long-range relations remain unresolved when mentions never co-occur within overlapping chunks. To address this, we perform a final document-level cluster linking step based on aggregated co-occurrence statistics between proper nouns. During chunk processing, we construct a proper-noun co-occurance matrix. For each proper noun p, we maintain a score for every other proper noun q that appears in the same chunk as p. This score is incremented when p and q are assigned to the same predicted cluster and decremented when they are assigned to different

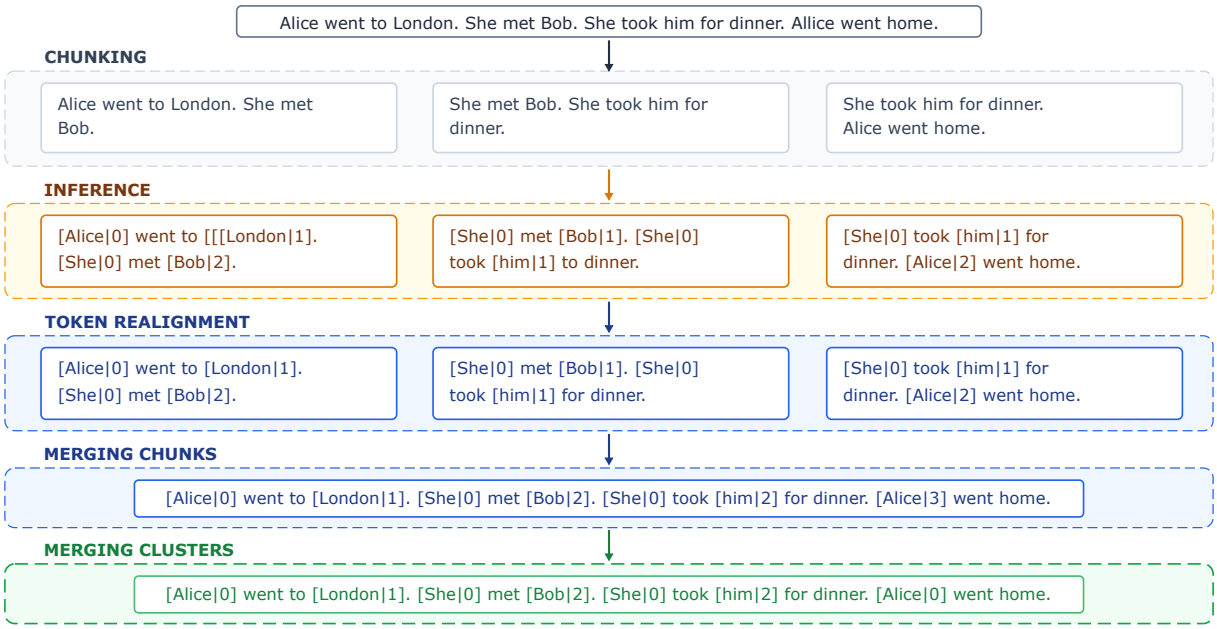

Figure 2: Coreference Resolution Pipeline

clusters. Scores are accumulated across all chunks, producing both positive and negative evidence for cross-cluster merging. For example, a strongly negative score indicates that two mentions frequently co-occur but are consistently assigned to different clusters, whereas a positive score reflects repeated coreference linking across multiple chunks.

After producing a merged document, we extract all predicted coreference clusters and retain only proper-noun mentions, removing pronouns and noun phrases. Clusters are then merged according to two constraints. Let C_i and C_j denote two clusters represented as sets of proper nouns:

$$C_i \text{ and } C_j \text{ are merged if}$$
$$C_i \cap C_j \neq \emptyset \wedge \forall p \in C_i, \forall q \in C_j,\ s(p,q) \geq 0.$$

The clusters must share at least one proper-noun mention. Merging is permitted only if no mention in C_i has a negative compatibility score with any mention in C_j. Again, Figure 2 shows an example. In the final document, cluster 0 can be merged with cluster 3.

4 Experiments

4.1 Datasets

We evaluated our approach using two literary coreference datasets, Litbank and MovieCoref.

The Litbank corpus is made up of 100 works of fiction (novels and short stories) written between 1719 and 1922. The corpus doesn't contain full texts, approximately 2000 words are selected from each work. Coreference annotations in Litbank cover six ACE entity categories (people, facilities, locations, geopolitical entities, organizations and vehicles). Litbank was chosen as it is the most widely used literary coreference benchmark. There are ten official splits [2]. The official metric is the cross-validation score across all splits. However, many previous researchers only used the first split. We use both the first split and the full ten-fold cross validation in our experiments.

MovieCoref is another small dataset. It consists of six full length screenplays. Although less widely used than LitBank, it provides a good contrast as it features full length documents. Previous evaluations use cross-validation over the six screenplays. We adopt the same approach in our experiments.

Statistics for the both datasets are shown in Table 1.

Table 1: Data statistics for the datasets

Dataset	# Docs	Avg. Words / Doc	Avg. Mentions / Doc	Avg. Cluster Size
LitBank	100	2,105	291	3.7
MovieCoref	6	22,434	3,831	73.2

[2] https://github.com/dbamman/lrec2020-coref/

4.2 Implementation

For fine-tuning we utlised the Unsloth AI library [3] to implement PEFT through Quantized Low-Rank Adaptation (QLoRA) (Dettmers et al., 2023). This memory efficient approach allowed us to train large models on a single A100 40GB GPU. We used a number of instruction-tuned base models in our experiments: Llama 3.1 (Team, 2024), Mistral Nemo (Mistral AI and NVIDIA, 2024), Gemma 3 (Gemma Team et al., 2025), Qwen 3 (Qwen Team et al., 2025) and Phi-4 (Microsoft Research, 2024).

As described in Section 3, input documents are processed in chunks. We used a chunk size of 1400 tokens. This chunk size was calculated based on models having a maximum pretraining context window of 4096 tokens, an expansion factor of 1.75 for additional tokens generated as part of the annotations, and additional tokens for the system prompt.

Models were fine-tuned, targeting all linear layers for low-rank adaptation. We used standard hyperparameters for training, but adopted aggressive settings for learning rate (2×10^{-4}) and the LoRA rank ($r = 32$) due to the complexity of coreference resolution as a task. We trained for a single epoch; a complete list of training configurations is provided in the appendix.

Once fined-tuned, we used the models as part of the pipeline shown in Figure 2

5 Results

We use the CorefUD scorer (Novák et al. (2024)) to report the CoNLL F1 score in each of our experiments. The CoNLL score is average of MUC, B3 and CEAFe.

5.1 Model Comparison

We compare the models shown in Table 2 using the first split of LitBank. Performance across the large models (8B to 15B parameters) remains relatively flat, with CoNLL F1 scores in a narrow range between 79.93 and 81.23. Llama 3.1 8B achieves the highest overall score of 81.23, outperforming larger architectures such as Qwen 3 and Phi 4. Understandably performance drops for the smaller models. However, even these smaller these models show capable performance, Gemma 3 1B achieves 71.5 and Phi 4 mini is able to score 73.45. Although we ran our experiments on an A100 processor, the memory requirements for these smaller models would comfortable run under the 16GB constraints of the widely available T4 processor and therefore present a viable alternative for accessible deployment.

Table 2: Comparison of various models on the first split of the Litbank test set. The score given is the CoNLL F1 score.

Model	Parameters (Billion)	LitBank_0
Phi 4 (instruct)	15	80.3
Qwen 3	14	79.93
Gemma 3	12	80.01
Mistral NeMo	12	80.06
Llama 3.1	8	81.23
Phi 4 mini	4	73.45
Gemma 3	1	71.5

5.2 Litbank

Table 3: Comparison of our fine-tuned Llama 3.1 8B model to prior work on the LitBank dataset (LitBank_0= split 1, LitBank=ten-fold cross-validation)

Model	LitBank_0	LitBank
Toshniwal et al. (2021)	77.20	79.30
Toshniwal et al. (2021) (joint)	78.20	—
Zhang et al. (2023)	77.30	78.30
Zhang et al. (2023) (joint)	81.20	—
Llama 3.1 8B + Our coref pipeline	81.23	81.32

Table 3 shows results of our coreference pipeline comparing the best performing model from Section 5.1 (Llama 3.1 8B) to previous work. For this experiment we evaluated using the remaining nine splits of LitBank to provide a 10-fold cross-validation score. We compare our results to the state-of-the-art (SOTA) findings presented by Toshniwal et al. (2021) and Zhang et al. (2023). These previous works reported results twice, once from training using only the Litbank dataset, and again using a joint dataset of OntoNotes, Preco and LitBank (indicated by the word 'joint' in Table 3). Toshniwal et al. (2021) used a neural model architecture designed specifically for coreference resolution. Zhang et al. (2023) used a fine-tuned 3 billion parameter T5 seq2seq model.

Our approach achieves excellent results, outperforming all previous work. On the full 10-fold cross-validation, our pipeline achieves an F1 score of 81.32. On LitBank split 0, our approach outperforms the single data versions of both Toshniwal et al. (2021) and Zhang et al. (2023), and even

[3] https://unsloth.ai/

narrowly exceeds the performance of the jointly trained model of Zhang et al. (2023).

These results back up the findings of Zhang et al. (2023) and Hicke and Mimno (2024) that SOTA results can be achieved simply by fine tuning off-the shelf models, without the need for a task specific architecture.

It is also important to note that we are able to achieve these results without requiring massive out-of-domain additional training datasets.

Perhaps the results aren't surprising, given we have used larger models than the one used by Zhang et al. (2023). However, we employ PEFT; only training a small percentage of parameters.

5.3 MovieCoref

Table 4: Comparison of our fine-tuned Llama 3.1 8B model to prior work on the MovieCoref dataset

Model	MovieCoref (F1)
Baruah and Narayanan (2023)	71.10
Llama 3.1 8B + Our coref pipeline	71.97

MovieCoref has received less attention that the LitBank dataset; as such, there is limited prior work. We compare our approach only to Baruah and Narayanan (2023). This work used Dobrovolskii (2021)'s word level neural coreference resolution model. Baruah and Narayanan (2023) pre-trained the model using the OntoNotes dataset before training with MovieCoref. We take the same evaluation approach of cross-validation using the six full length screenplays. Again, our pipeline is able to achieve a high level of accuracy (71.97) which again outperforms the previous SOTA results. This is particularly impressive as the model was trained using a very small number of examples; we didn't have the advantage of pre-training on an additional dataset. Despite the challenges of a small dataset and performing coreference resolution across full documents, a score of 71.97 isn't far from the the LitBank results, and indicates the utility of our approach for processing complete documents. We attribute this score partially to the effectiveness of our pipeline's merging clusters action, but also the inherent characteristics of screenplays. Screenplays are structured documents with clear conventions for indicating dialogue, speakers and actions. Unlike novels which can feature complex prose, descriptions in screenplays are direct, unambiguous instructions for filmmakers. This structural clarity makes it easier for the model to recognise and resolve links between entities.

It is also worth noting that the MovieCoref dataset is not only limited in size but it is noisy: featuring inconsistent use of whitespace, page numbers and spelling mistakes. With a cleaned, larger dataset we would expect our results to be even higher.

5.4 Ablation study

We consider how effective the cluster linking approach we outlined in Section 3 is in resolving long distance coreference links. Again we used the Llama 3.1 model. *No merge clusters* indicates the pipeline was executed as in Figure 2 but without this final merge clusters action. We evaluated over the full cross validation results for both LitBank and MovieCoref. The merge clusters action makes little difference for the LitBank dataset, a 0.3 reduction in CoNLL F1 score. However, in MovieCoref, it makes a significant difference (18.79). This is understandable, the LitBank documents are excerpts so don't have long distances between mentions. The results for MovieCoref demonstrate, that although very simple, our approach to merging clusters on the basis of proper nouns is effective.

Table 5: Ablation study for Merging Clusters

	LitBank	MovieCoref
Full coref pipeline	81.32	71.97
No merge clusters	-0.3	-18.79

6 Conclusion

As part of our wider pipeline, we have shown that fine-tuned decoder-only causal language models can achieve state-of-the-art performance for literary coreference resolution. We demonstrated a memory efficient approach built around a QLoRa implementation of PEFT that allowed us run training efficiently using a single GPU. Our approach works effectively across the LitBank dataset of novels and the MovieCoref screenplays dataset. Importantly, this includes full length literary works. In the future we plan to use a similar approach for related information extraction tasks e.g. event detection and semantic role labelling. In doing so we aim to provide solid groundwork for advancements in computational literary analysis, for example, in

modelling narrative structures and character networks.

7 Limitations

There are a number of limitations in the work we have presented here.

- Our approach to merging clusters is basic, relying on the same proper noun to be used in disparate clusters. Although this works for many literary texts, this won't robustly work across all literary works. For example, in Cormac McCarthy's The Road, the two main characters are only ever referred to by noun phrases, 'the man' and 'the boy'.

- Related to the above point, the scope of our evaluation is constrained by the small scale of the available benchmarks. Both LitBank (consisting of only 100 literary excerpts) and MovieCoref (comprising just six screenplays) represent limited sample sizes. Consequently, our results cannot be definitively generalized to all literary works.

- Our approach relies on the model replicating the entire input sequence with the addition of coreference annotations. This verbose output limits scalability. To avoid exceeding the original pretraining context window of some models, our chunk size is limited. Alternative approaches, for example, simply outputting the coreferring mentions would result in shorter model outputs and allow larger chunks sizes to be processed.

References

Apoorv Agarwal, Anup Kotalwar, and Owen Rambow. 2013. Automatic extraction of social networks from literary text: A case study on alice in wonderland. In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pages 1202–1208. Asian Federation of Natural Language Processing / ACL.

David Bamman, Sejal Popat, and Sheng Shen. 2019. An annotated dataset of literary entities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2138–2144. Association for Computational Linguistics.

Sabyasachee Baruah and Shrikanth Narayanan. 2023. Character coreference resolution in movie screenplays. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10300–10313. Association for Computational Linguistics.

Bernd Bohnet, Chris Alberti, and Michael Collins. 2023. Coreference resolution through a seq2seq transition-based system. *Trans. Assoc. Comput. Linguistics*, 11:212–226.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Vladimir Dobrovolskii. 2021. Word-level coreference resolution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7670–7675. Association for Computational Linguistics.

Yufeng Du, Minyang Tian, S. Ronanki, Subendhu Rongali, S. Bodapati, A.G. Galstyan, Azton Wells, Roy Schwartz, Eliu A. Huerta, and Hao Peng. 2025. Context length alone hurts llm performance despite perfect retrieval. *ArXiv*, abs/2510.05381.

Gemma Team, CJ Choquette-Choo, Carey Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, and 1 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Philip Gorinski and Mirella Lapata. 2015. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076.

Rebecca M. M. Hicke and David Mimno. 2024. [lions: 1] and [tigers: 2] and [bears: 3], oh my! literary coreference annotation with llms. *CoRR*, abs/2401.17922.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguistics*, 8:64–77.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel S. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 5802–5807. Association for Computational Linguistics.

Yuval Kirstain, Ori Ram, and Omer Levy. 2021. Coreference resolution without span representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 14–19. Association for Computational Linguistics.

Vincent Labatut and Xavier Bost. 2019. Extraction and analysis of fictional character networks: A survey. *ACM Comput. Surv.*, 52(5):89:1–89:40.

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, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 188–197. Association for Computational Linguistics.

Microsoft Research. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Mistral AI and NVIDIA. 2024. Mistral nemo 12b open-weights foundation model. `https://huggingface.co/mistralai/Mistral-Nemo-Instruct-2407`. Accessed: 2026-05-20.

Eric T Nalisnick and Henry S Baird. 2013. Character-to-character sentiment analysis in shakespeare's plays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 479–483.

Michal Novák, Barbora Dohnalová, Miloslav Konopik, Anna Nedoluzhko, Martin Popel, Ondrej Prazak, Jakub Sido, Milan Straka, Zdeněk Žabokrtský, and Daniel Zeman. 2024. Findings of the third shared task on multilingual coreference resolution. In *Proceedings of The Seventh Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 78–96, Miami. Association for Computational Linguistics.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Qwen Team, Jinze Bai, Shuai Deng, Xiaodong Fan, Zhe Gan, Binyuan Gong, Junyang Lin, An Yang, Chang Zhou, and Jingren Zhou. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Ina Rösiger, Sarah Schulz, and Nils Reiter. 2018. Towards coreference for literary text: Analyzing domain-specific phenomena. In *LaTeCH@COLING*.

Shashank Srivastava, Snigdha Chaturvedi, and Tom M. Mitchell. 2016. Inferring interpersonal relations in narrative summaries. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2807–2813. AAAI Press.

Llama Team. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

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, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3921–3931. 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 2020, Online, November 16-20, 2020*, pages 8519–8526. Association for Computational Linguistics.

Shubham Toshniwal, Patrick Xia, Sam Wiseman, Karen Livescu, and Kevin Gimpel. 2021. On generalization in coreference resolution. *CoRR*, abs/2109.09667.

Ted Underwood, David Bamman, and Sabrina Lee. 2018. The transformation of gender in english-language fiction. *Journal of Cultural Analytics*, 3(2).

Gorka Urbizu, Ander Soraluze, and Olatz Arregi. 2020. Sequence to sequence coreference resolution. In *Proceedings of the Third Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 39–46, Barcelona, Spain (online). Association for Computational Linguistics.

Hardik Vala, David Jurgens, Andrew Piper, and Derek Ruths. 2015. Mr. bennet, his coachman, and the archbishop walk into a bar but only one of them gets recognized: On the difficulty of detecting characters in literary texts. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 769–774. The Association for Computational Linguistics.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. Corefqa: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6953–6963. Association for Computational Linguistics.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential reasoning learning for language representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7170–7186. 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, EMNLP 2023, Singapore, December 6-10, 2023*, pages 11493–11504. Association for Computational Linguistics.

A System Prompts

Listing 1: System Prompt for the LitBank Dataset

```
You are an expert coreference resolution assistant.

Annotate coreferring mentions using the format:

[span|id]

Follow these rules:
- Annotate people, facilities, locations,
    geopolitical entities, organizations and
    vehicles.
- Include names, nominals, and pronouns if they
    refer to the same entity.
- Annotate the maximal extent of a span.
- Only annotate mentions that participate in a
    coreference chain.
- Do not annotate events, predicates, or verb
    phrases.
- Use the same id for all mentions that corefer.
- Assign new ids for each new discourse entity.
- Output ONLY the annotated text in the required
    bracket format. Do not output any other text.
```

Listing 2: System Prompt for the MovieCoref Dataset

```
You are an expert coreference resolution assistant.

Annotate coreferring mentions using the format:[span
    |id]

Follow these rules:
- Annotate only characters and 'we' when it
    references the reader or camera's point of view
    .
- Include names, nominals, and pronouns if they
    refer to the same entity.
- Annotate the maximal extent of a span.
- Only annotate mentions that participate in a
    coreference chain.
- Do not annotate events, predicates, or verb
    phrases.
- Use the same id for all mentions that corefer.
- Assign new ids for each new discourse entity.
- Output ONLY the annotated text in the required
    bracket format. Do not output any other text.
```

B Training Parameters

Hyperparameter	Value
LoRA Rank (r)	32
LoRA Alpha (α)	64
LoRA Dropout	0.0
Bias Type	none
Rank-Stabilised LoRA (`use_rslora`)	True

Table 6: Parameter configurations for Low-Rank Adaptation (LoRA) fine-tuning.

Hyperparameter	Value
Learning Rate	2×10^{-4}
LR Scheduler Type	linear
Warmup Ratio	0.1
Weight Decay	0.01
Num Train Epochs	1
Max Sequence Length	4,096
Per-Device Train Batch Size	1
Per-Device Eval Batch Size	1
Gradient Accumulation Steps	4
Gradient Checkpointing	True

Table 7: Hyperparameters used for Supervised Fine-Tuning (SFT).

C Additional Results

Cross-val Split	CoNLL F1
0	81.23
1	82.20
2	80.70
3	81.18
4	80.89
5	79.00
6	83.44
7	80.68
8	84.47
9	79.36
Mean Average	**81.315**

Table 8: Full Cross-validation results for Litbank and Llama 3.1 8B

Cross-val Split	CoNLL F1
0	69.99
1	73.86
2	78.42
3	73.60
4	65.99
5	69.98
Mean Average	**71.973**

Table 9: Full Cross-validation results for MovieCoref and Llama 3.1 8B