xCoRe: Cross-context Coreference Resolution

Giuliano Martinelli[1], **Bruno Gatti**[1], **Roberto Navigli**[1,2]
[1]Sapienza NLP Group, Sapienza University of Rome
[2]Babelscape
{martinelli, gatti, navigli}@diag.uniroma1.it

Abstract

Current coreference resolution systems are typically tailored for short- or medium-sized texts and struggle to scale to very long documents due to architectural limitations and implied memory costs. However, a few available solutions can be applied by inputting documents split into smaller windows. This is inherently similar to what happens in the cross-document setting, in which systems infer coreference relations between mentions that are found in separate documents.

In this paper, we unify these two challenging settings under the general framework of cross-context coreference, and introduce xCoRe, a new unified approach designed to efficiently handle short-, long-, and cross-document coreference resolution. xCoRe adopts a three-step pipeline that first identifies mentions, then creates clusters within individual contexts, and finally merges clusters across contexts. In our experiments, we show that our formulation enables joint training on shared long- and cross-document resources, increasing data availability and particularly benefiting the challenging cross-document task. Our model achieves new state-of-the-art results on cross-document benchmarks and strong performance on long-document data, while retaining top-tier results on traditional datasets, positioning it as a robust, versatile solution that can be applied across all end-to-end coreference settings. We release our models and code at http://github.com/sapienzanlp/xcore.

1 Introduction

Coreference resolution (CR) is a Natural Language Processing task that aims to identify and group mentions that refer to the same entity (Karttunen, 1969). Although modern neural models have reached near-human performance on standard document-level benchmarks such as OntoNotes (Pradhan et al., 2012) and PreCo (Chen et al., 2018), coreference resolution remains far from solved in two challenging settings: i) coreference on very long documents, for which models require maintaining coherence on extended inputs, and ii) cross-document coreference, which requires resolving entity relations across multiple documents.

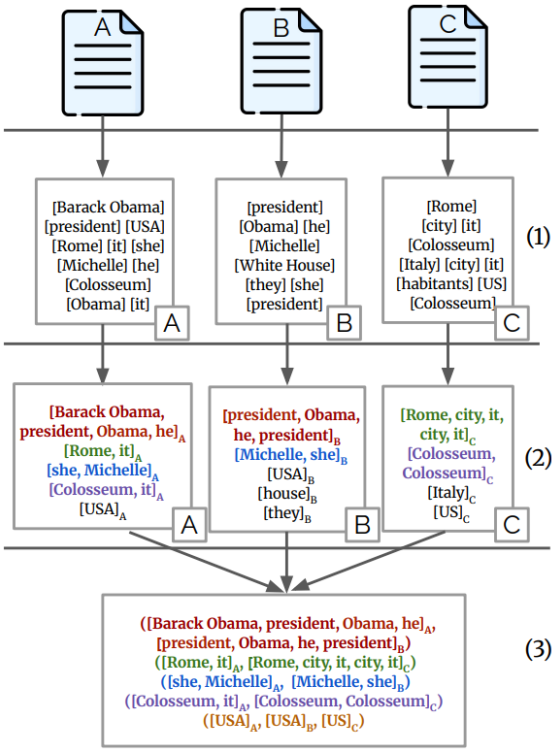

Figure 1: The xCoRe pipeline: for each input context we adopt: (1) within-context mention extraction, to extract possible mentions, (2) within-context mention clustering, to build local clusters, and (3) cross-context cluster merging, to obtain the final set of cross-context clusters.

Most available coreference techniques are typically tailored to short- to medium-sized documents and struggle to process longer inputs due to the quadratic complexity of their underlying Transformer-based architectures. To address this problem, recent solutions have proposed segment-

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 34264–34278

ing long documents and processing them independently (Toshniwal et al., 2021; Guo et al., 2023; Liu et al., 2025), a method that inevitably trades efficiency at the cost of lowering performance (Gupta et al., 2024). A similar problem occurs in the cross-document setting, where state-of-the-art techniques that separately encode texts cannot surpass 35 CoNLL-F1 points (Cattan et al., 2021a) on the ECB+ benchmark (Cybulska and Vossen, 2014).

In these coreference scenarios, which have always been treated as two distinct settings, current architectures suffer from a shared limitation: models struggle to resolve coreference across disjoint contexts. In this paper, we frame this general problem as cross-context coreference resolution, and propose xCoRe, a new end-to-end neural architecture designed for every coreference scenario. xCoRe operates in three stages: (1) within-context mention extraction, (2) within-context mention clustering, and (3) cross-context cluster merging. Our pipeline, shown in Figure 1, is inspired by the observation that existing models perform well within single documents; our approach builds on this foundation by learning to merge local clusters across different contexts.

In our experiments, we demonstrate that our new general cross-context formulation is particularly beneficial because it enables training across shared long- and cross-document resources, increasing data availability and improving model performance. We extensively evaluate xCoRe on a suite of long-document, cross-document, and traditional coreference datasets, demonstrating its overall robustness and flexibility across settings and obtaining new state-of-the-art scores for end-to-end coreference resolution on every cross-document benchmark and top-tier results on long-document benchmarks.

2 Related Work

In this Section, we review recent approaches to long- and cross-document coreference resolution. We discuss the limitations of existing models to scale from medium-sized documents to significantly longer or multiple documents, highlighting the core challenge of cross-context coreference.

2.1 Long-Document Coreference Resolution

Resources Coreference resolution performance is usually evaluated on medium-sized texts such as the OntoNotes benchmark (Pradhan et al., 2012), with around 450 tokens per document. However, recent work has focused on evaluating the performance of models on longer texts, such as in LitBank (Bamman et al., 2020), an annotated benchmark of 100 literary text samples from the literature genre. The main limitation of LitBank is that it truncates book samples to 2,000 tokens and does not capture coreference relations that are found across entire books. We also consider two full book resources that have been introduced recently: i) The Animal Farm narrative book, manually annotated by Guo et al. (2023), and ii) BookCoref (Martinelli et al., 2025), a new full-book coreference resolution benchmark with silver-annotated training set and gold annotated test set.

Long-document systems The lack of very long manually-annotated documents has caused state-of-the-art coreference resolution techniques to focus only on short- or medium-sized sequences, adopting techniques that cannot be applied to longer texts such as books or long newspaper articles. Among such approaches, generative models are currently impractical for processing very long texts, since they require the entirety of the input text to be regenerated, doubling the context length. This is unfeasible for long document settings, since these approaches rely on memory-demanding Transformer architectures. These concerns scale to Large Language Models (LLMs) too, and, although their applicability in the CR task is still under discussion, current methods for LLM-based CR have yet to reach the performance of fine-tuned encoder-only models (Le and Ritter, 2023; Porada et al., 2024).

In contrast, discriminative encoder-only models are more suited for processing longer sequences, being more memory- and time-efficient. Among models adaptable to longer inputs, Maverick (Martinelli et al., 2024) is an optimal choice, since it combines state-of-the-art scores on LitBank and OntoNotes, and it can theoretically handle up to 25,000 tokens. However, its self-attention mechanism makes it practically unusable on very long documents because of quadratic memory costs. This is solved in specially tailored solutions for long-document coreference, such as Longdoc (Toshniwal et al., 2020, 2021) and Dual-cache (Guo et al., 2023), which encode full documents in smaller windows and incrementally build coreference clusters by dynamically "forgetting" less relevant entities via a global cache of recently predicted mentions.

Another recent approach for long documents is

presented in Gupta et al. (2024), a method that hierarchically merges clusters from smaller windows of long documents, performing several pairwise cluster merging steps. However, its effectiveness has only been evaluated on German texts, and it exhibits several limitations: it cannot handle singleton mentions, requires separate training for the hierarchical merging module, and involves multiple merging stages to compute the final document-level clusters. In our work, we address these problems by proposing a modular, end-to-end architecture designed for cross-context coreference resolution, which performs cluster merging in a single pass and eliminates the need for multi-stage or separately trained merging components.

2.2 Cross-document Coreference Resolution

We now review related cross-document works, focusing on traditional entity-based coreference works and not including the identification and linking of events. Moreover, to align with standard practice in the traditional and long-document coreference settings, we specifically focus on end-to-end coreference resolution, usually referred to as "using predicted mentions" (Cattan et al., 2021a). We therefore do not report techniques that need to start from gold mentions, as they require additional resources that prevent them from being applied to realistic applications (Cattan et al., 2021a) and fall outside the focus of our work.

Resources The most widely used dataset for cross-document CR is ECB+ (Cybulska and Vossen, 2014), which contains 996 news articles grouped into 43 sets of documents, each of which represents a topic. Notably, both events and entities are annotated in ECB+, and entities are annotated only if they participate in events. A more recent dataset, SciCo (Cattan et al., 2021c), focuses on scientific documents. It is approximately three times larger than ECB+ and includes annotations for entities only, drawn from segments of scientific papers. Recent efforts to evaluate LLMs on ECB+ and SciCo include the SEAM benchmark (Lior et al., 2024), which shows that, even with long context lengths and access to gold mentions, LLMs perform poorly on cross-document CR tasks.

Cross-document models Most existing models for cross-document coreference assume access to gold mentions. Among them, Cross Document Language Modeling (Caciularu et al., 2021, CDLM) currently achieves the best performance on ECB+. It employs Longformer (Beltagy et al., 2020) as a cross-encoder and processes each pair of sentences in a topic separately. However, this results in significant computational overhead since both time and memory complexity are quadratic with the number of sentences (Hsu and Horwood, 2022). More importantly, CDLM requires gold mentions, making it impractical for end-to-end applications starting from raw text.

To address this, Cattan et al. (2021a) propose an architecture for cross-document CR that starts from predicted mentions. Their system builds upon the end-to-end coreference pipeline of Lee et al. (2017), which includes mention extraction followed by mention clustering, and extends it to handle multiple documents. Their traditional mention-to-mention approach requires separate training for the mention extractor and the clustering module, along with the tuning of several hyperparameters for mention pruning. In our work, we eliminate the need for handcrafted features, separate modules, or threshold tuning, providing a practical solution that builds cross-document predictions from locally extracted clusters.

3 Methodology

We now present xCoRe, a unified coreference system capable of seamlessly handling short-, long-, and multi-document inputs. We first present our cross-context formulation in Section 3.1. Then, in Section 3.2, we present the xCoRe three-step discriminative pipeline, which first constructs coreference clusters within local contexts and then merges them across contexts in a single forward pass. Finally, in Section 3.3, we detail our training and inference strategies.

3.1 Cross-context Formulation

We define cross-context coreference as the general task of inferring coreference relations between mentions that are found in distinct chunks of text, which we refer to as *contexts*. With xCoRe, we propose a novel architecture, training, and inference strategy for cross-context coreference scenarios. Our general approach can handle any set of generic contexts $c_1, c_2, \ldots, c_n \in C$ and can naturally be applied to the cross-document setting by processing its documents separately. When dealing with short documents, our pipeline is applied to a single context and handles this base case by executing only the first two local steps of the xCoRe pipeline,

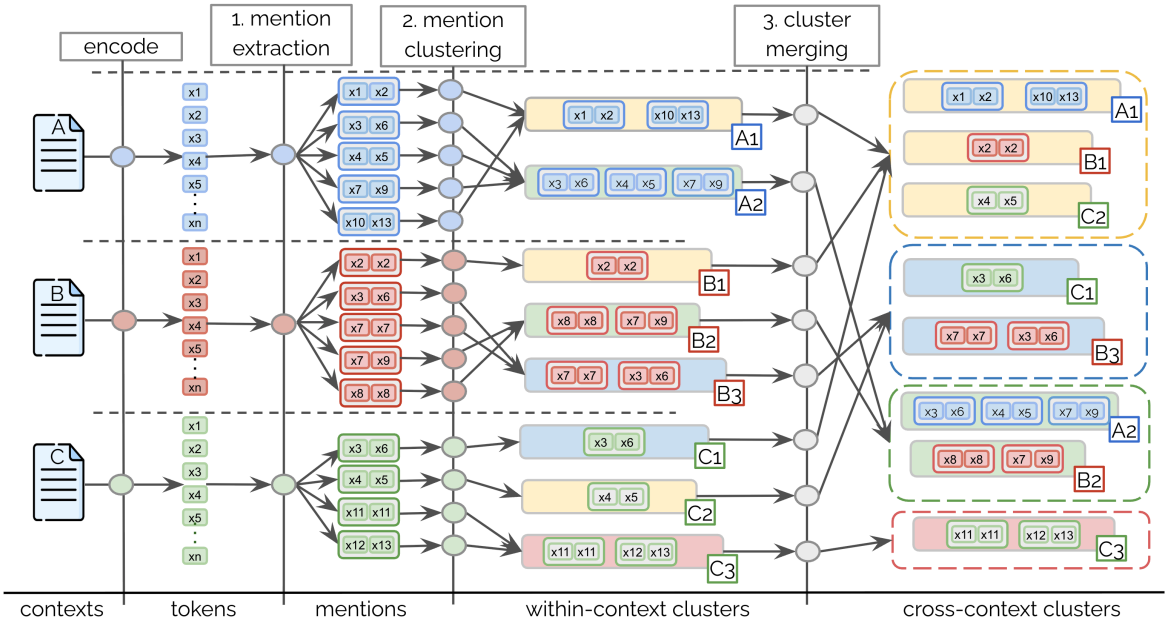

Figure 2: Illustration of the xCoRe architecture, which takes as input multiple contexts, illustrated as "A", "B", and "C", and outputs their merged coreference clusters. For each context, within-context clusters are extracted via within-context (1) mention extraction and (2) mention clustering. Finally, the cross-context (3) cluster merging step is applied to form clusters at the cross-context level.

i.e., mention extraction and mention clustering. However, when a single document exceeds a certain length, determined by available memory constraints, it is divided into multiple fixed-size contexts[1], in which every $c_i \in C$ is a single fixed-length window.

3.2 Model Architecture

We now introduce our model pipeline, detailed in Figure 2. In xCoRe , the first within-context mention extraction and clustering steps of our pipeline are built upon the traditional mention–antecedent approach introduced by Lee et al. (2017, 2018), where the most probable mentions are first identified and then linked to their most likely coreferent mentions. However, the main innovation of xCoRe lies in its cluster merging strategy, which enables the formation of coherent clusters across multiple text windows with a simple yet effective technique: for each cluster identified within independent contexts, the model learns to predict its most likely cross-context match.

3.2.1 Within-Context Coreference Resolution

In xCoRe, we first perform a within-context coreference resolution step for each context $c_i \in C$ in the input. This step is divided into within-context mention extraction, which deals with the extraction of all possible mentions in the input context, and within-context mention clustering, which aims to find the most probable coreferring mentions for all the previously extracted mentions.

Since this step is based on well-established methods and serves as a stepping stone for our new cluster merging strategy, we provide a short overview of our within-context methodology here, and leave a detailed discussion of it to Appendix A.

Mention Extraction To extract mentions from each context $c_i \in C$, we adopt an equivalent approach to Maverick (Martinelli et al., 2024), the latest advancement in discriminative encoder-only models. Specifically, we adopt the start-to-end mention extraction strategy in which we first identify all the possible starts of a mention, and then, for each start, extract its possible end. Formally, we first compute the hidden representation $(x_1^{c_i}, \ldots, x_n^{c_i})$ of the tokens $(t_1^{c_i}, \ldots, t_n^{c_i}) \in c_i$ using a Transformer-based encoder. For all the tokens that have been predicted as the start of a mention, i.e., $t_s^{c_i}$, we then predict whether its subsequent tokens $t_j^{c_i}$, with $s \leq j$, are the end of a mention that starts with $t_s^{c_i}$. In this process, we use an end-of-sentence mention regularization strategy: after extracting a possible start, we only consider its pos-

[1]Window size is set to its maximum based on hardware constraints to take advantage of the well-known high performance of within-document coreference.

sible tokens up to the nearest end-of-sentence. At the end of this step, we end up with a final set of possible mentions M^{c_i} for each $c_i \in C$.

Mention Clustering After extracting all the possible mentions $m_j^{c_i} \in M^{c_i}$ from c_i, we use a mention clustering strategy based on LingMess (Otmazgin et al., 2023) and adopted in Maverick. Specifically, for each mention $m_j^{c_i} = (x_s^{c_i}, x_e^{c_i})$ and antecedent mention $m_k^{c_i} = (x_{s'}^{c_i}, x_{e'}^{c_i})$, each represented as the concatenation of their respective start and end token hidden states, we use a set of linear classification layers to detect whether $m_k^{c_i}$ is coreferring with $m_j^{c_i}$. Notably, after these within-context steps, as illustrated in Figure 2, for each context c_i provided in input, we can extract its coreference clusters $\mathcal{W}^{c_i} = \{\mathcal{W}_1^{c_i}, \mathcal{W}_2^{c_i}, \ldots, \mathcal{W}_m^{c_i}\}$, with $\mathcal{W}_j^{c_i} = (m_{j_1}^{c_i}, \ldots, m_{j_z}^{c_i})$, that subsequently will be merged in the cluster merging step of the pipeline.

3.2.2 Cross-context Cluster Merging

This step is our new key component to produce the final cross-context coreference clusters by merging local clusters. While all the previous steps are applied to single contexts and are executed sequentially, this step starts after all the within-context clusters $\mathcal{W}^{c_i}$ have been extracted across all contexts $c_i \in C$. We first compute the representation for each cluster $\mathcal{W}_j^{c_i} \in \mathcal{W}^{c_i}$ in all the contexts $c_i \in C$ obtained in the previous step, using a single-layer Transformer T to encode the hidden states of each of its mentions as:

$$hs(\mathcal{W}_j^{c_i}) = \mathrm{T}(m_{j_1}^{c_i}, \ldots, m_{j_z}^{c_i}).$$

After this, we compute the pairwise coreference probability p_{cm} between clusters' hidden representations using a linear classification layer as:

$$\mathcal{L}(x) = W \cdot (ReLU(W' \cdot x))$$

$$p_{cm}(\mathcal{W}_a^{c_i}, \mathcal{W}_b^{c_j}) = \mathcal{L}(hs(\mathcal{W}_a^{c_j}) \| hs(\mathcal{W}_b^{c_i}))$$

where W, W' are learnable parameters, c_i, c_j are arbitrary contexts and $\mathcal{W}_b^{c_i}$, $\mathcal{W}_a^{c_j}$ are two arbitrary coreference clusters in c_i and c_j. We calculate this probability and take the most probable coreferent cluster for every pair of clusters $\mathcal{W}_a^{c_i}, \mathcal{W}_b^{c_j}$ from $c_i, c_j \in C$ respectively, with $c_i \neq c_j$. We do not compare clusters that come from the same context, i.e., $c_i = c_j$, since they have been predicted separately by the previous cluster merging step, and take the most probable coreferent cluster for each pair of mentions with $p_{cm} > 0.5$, leaving the cluster as a singleton when none of the others are predicted coreferential. Notably, this technique is invariant to the order of cluster appearance, and is therefore applicable both when contexts have a sequential order, such as in long documents, and when they are not ordered, as in cross-document settings. As a result of this step, by sequentially merging coreferential clusters, we obtain a final set of cross-context clusters.

3.3 Cross-context Training and Inference

At inference time, as reported in Section 3.1, we address the quadratic memory complexity of encoding long sequences by splitting long documents into fixed-size windows c_i of maximum possible context length w. Similarly, when dealing with multiple documents, each text is encoded as a separate context c_i. Nevertheless, in this scenario, training models adopting a traditional supervised fine-tuning technique presents a unique challenge: to effectively learn cross-context cluster merging, during training, the model must be exposed to training examples containing multiple contexts. For this reason, one of our training objectives is to build training batches in which our model can learn to deal with a large number of contexts. On the other hand, since we also want our model to be reliable in the within-context coreference step, it is crucial to train on samples of long individual contexts. These two training objectives cannot easily be fulfilled together, since encoding many long contexts would inherently imply a significant memory overhead.

We address this problem by designing a dynamic batching training strategy. When dealing with single-document datasets, we train on contiguous contexts extracted from the original training documents $d_i \in D$, choosing a different number and dimension of input contexts at each training step. Specifically, at each step, we first sample the number of training contexts n in the range $(1, \lfloor w/s \rfloor)$, where w is the previously detailed maximum context length, and s is the average sentence length of our dataset. Then, we construct a training batch by sampling n continuous contexts from d_i, with length equal to $\frac{min(w, |d_i|)}{n}$ and round up context boundaries to the nearest end of sentence. When dealing with cross-document datasets, we use an analogous approach: n is chosen in the range $(1, \lfloor w/dl \rfloor)$, with dl being the average document length of our training dataset. In this case,

Dataset	Type	Topics	Train	Dev	Test	Tokens	Mentions	Singletons
ECB+	cross-document	43	594	196	206	107k	8289	1431
SciCo	cross-document	521	9013	4120	8237	2.1M	26222	2721
Animal Farm	long-document	-	-	-	1	35k	1705	0
LitBank	long-document	-	80	10	10	210k	29k	5742
BookCoref	long-document	-	45	5	3	11M	992k	0
PreCo	medium-size	-	36120	500	500	12.3M	3.9M	2M
OntoNotes	medium-size	-	2802	343	348	1.6M	194k	0

Table 1: Overview of the datasets used in our experiments across medium-size, long-, and cross-document coreference settings. For each dataset, we report the number of topics in cross-document datasets, the train/dev/test split sizes, and total number of tokens, annotated coreference mentions, and singleton mentions.

our training batch is built simply by collecting n documents from our training dataset.

This allows models to learn to deal both with inputs of many small contexts and with inputs of a few very large contexts, thereby fulfilling our two training objectives and allowing systems to be trained in constrained memory settings. We refer the reader to Appendix B.1 for a detailed description of our training strategy.

4 Experimental setup

4.1 Datasets

We now report technical details of the benchmarks adopted in the following sections, and refer the reader to Table 1 for dataset statistics.

In the cross-document setting, we train our models on the well-established ECB+ (Cybulska and Vossen, 2014) and SciCo (Cattan et al., 2021c) training sets, and test their results on the respective test sets. Specifically, to compare our results with previous work, in both datasets we test our models using gold topic information and excluding singleton mentions, since they have been shown to alter benchmark results (Cattan et al., 2021b). For the ECB+ dataset, we only deal with entity coreference resolution and do not include information from additional parts of the documents (usually referred to as the *Cybulska setting*, cf. Appendix C), differently from previous works that instead use additional surrounding context from the original documents contained in ECB+. Furthermore, in cross-document experiments, we follow previous work and input only documents that are within a single topic, leveraging the gold topic structure.

For long-document coreference, we train our comparison systems on the LitBank training data (Bamman et al., 2020) and on the silver-quality training set of BookCoref (Martinelli et al., 2025). The models trained on LitBank are tested on Animal Farm (Guo et al., 2023) and on the LitBank test set, while the models trained on BookCoref are tested on its manually-annotated test set. When testing on long documents, specifically on Animal Farm and BookCoref, in order to compare with previous work, we use a within-window size of w = 4000 tokens. Finally, we also include results on medium-size benchmarks such as OntoNotes (Pradhan et al., 2012) and PreCo (Chen et al., 2018).

4.2 Comparison Systems

We compare xCoRe performances against the current available systems for medium-size, long- and cross-document coreference.

Among models that are specifically tailored for cross-document coreference, we report the scores for the only available system that uses predicted mentions (Cattan et al., 2021c), which we refer to as PMCoref. Notably, since PMCoref uses additional document information when tested on ECB+ and has never been tested on SciCo, we replicate its results in order to be consistent with recent techniques and our xCoRe method. We also include the results of the current state-of-the-art technique, i.e., CDLM (Caciularu et al., 2021), which requires explicit gold mentions and is highly impractical owing to memory and time consumption. Additionally, we report the results shown in the recent work of Lior et al. (2024) in which they test Mistral-7B (Jiang et al., 2023) and LLamax3-70B (Grattafiori et al., 2024) on cross-document tasks.

Among systems for long-document coreference, we report the scores of two long-document incremental formulations, namely, Longdoc (Toshniwal et al., 2020) and Dual-cache (Guo et al., 2023). We also include Hierarchical-coref (Gupta et al., 2024), which builds long-document clusters using several hierarchical pairwise steps, and Maverick (Martinelli et al., 2024), which adopts the traditional mention-to-antecedent scoring strategy. Ad-

Models	*LitBank-Split* (CoNLL-F1)					*ECB+ Sampled* (CoNLL-F1)				
	Full	2 splits	4 splits	8 splits	20 splits	1 doc	2 docs	4 docs	8 docs	Full
xCoRe-append	78.2	72.4	57.3	39.8	27.1	55.7	29.8	22.8	14.2	11.8
xCoRe-m2a	78.0	76.4	75.8	73.0	70.3	54.8	40.8	39.1	36.9	35.1
xCoRe	**78.2**	**77.6**	**77.1**	**74.9**	**72.4**	**58.9**	**50.1**	**46.8**	**44.4**	**40.3**

Table 2: Results of xCoRe alternative merging strategies on *LitBank-Split* and *ECB+ Sampled*, in CoNLL-F1 points. To ensure robust results, ECB+ measurements are averaged using 10 different random samples of documents.

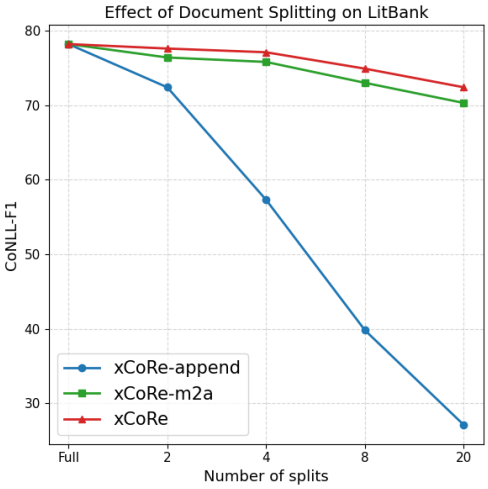

Figure 3: CoNLL-F1 scores comparison on LitBank with increasing number of splits per document.

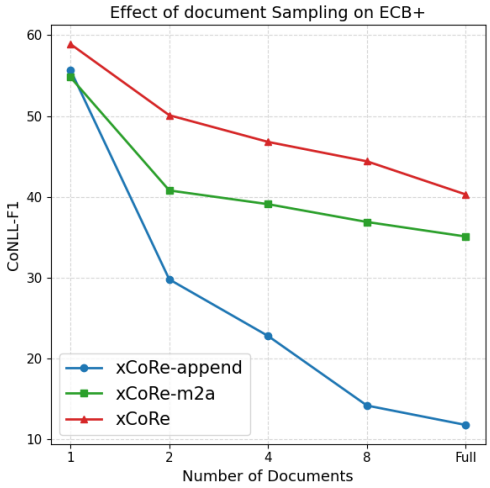

Figure 4: CoNLL-F1 scores comparison on ECB+ with increasing number of documents per topic.

ditionally, we include the system of Zhang et al. (2023, seq2seq), which uses a seq2seq methodology based on a very large generative model with 11 billion parameters. We exclude from our comparison systems the recent work of Zhu et al. (2025) because their results are computed on a different LitBank cross-validation setting, and their model was trained on 90 documents, including the validation split, which makes it not comparable to our reported systems. In Appendix C, we further detail our datasets, systems, and training setup.

4.3 xCoRe Systems

Pretrained Models Since our cross-context setting enables us to train systems on shared long- and cross-document resources, we also measure the benefits of pretraining xCoRe on datasets from different settings. Specifically, we report the performance of i) an xCoRe model pre-trained on LitBank (i.e. $\text{xCoRe}_{\text{LitBank}}$) on the cross-document setting, by additionally training and testing it on cross-document data, and ii) an xCoRe model pre-trained on ECB+ (i.e. $\text{xCoRe}_{\text{ECB+}}$) on the long-document setting, by additionally training and testing it on long-document data (see Section 4.1).

Cluster Merging Baselines To test the effectiveness of our new cluster merging strategy, we compare it against two baseline systems: i) xCoRe-*append*, in which cluster merging is disabled and within-context clusters are simply concatenated, and ii) xCoRe-*m2a*, which instead uses a traditional mention-to-antecedent strategy to compute cross-context clusters. Specifically, the only difference between xCoRe-*m2a* and a traditional mention-to-antecedent model applied on full documents (such as Maverick) is that contexts are encoded separately, and their hidden representations are not contextualized to the full document. Comparing xCoRe with these two settings shows i) whether our model can effectively learn the cluster merging task, and ii) whether it can surpass the traditional strategy of building clusters at the mention level.

5 Results

5.1 Cluster Merging Analysis

We first analyze the impact of the cluster merging approach, and report our results in Table 2 and in Figures 3 and 4. Specifically, we evaluate xCoRe, xCoRe-*append* and xCoRe-*m2a* on *LitBank-Split*,

Model	ECB+		SciCo	
	Pred.	Gold	Pred.	Gold
Baselines				
Mistral-7B	-	20.1	-	31.1
Llama-3x70B	-	22.3	-	24.4
CDLM	-	**82.9**[*]	-	**77.2**
PMCoref	35.7[*]	65.3[*]	-	66.8
PMCoref[†]	33.7	63.3	23.3	66.8
xCoRe (Ours)				
xCoRe	40.3	73.8	27.8	62.3
$\text{xCoRe}_{\text{LitBank}}$	**42.4**	74.1	**30.5**	67.3

Table 3: Results on ECB+ for comparison systems in terms of CoNLL-F1 score. We use (*) to indicate models that use additional context, (†) for replicated results without additional context, and (-) for results that were not reported in the original papers. Pred. and Gold indicate whether the model starts from predicted or gold mentions, respectively.

Model	Animal Farm	LitBank	BookCoref
Baselines			
Longdoc	25.8	77.2	**67.0**
Dual-cache	36.3	77.9	58.9
Hierarchical	27.9	61.5	42.8
seq2seq	-	77.3	-
Maverick	-	78.0	61.0
xCoRe (Ours)			
xCoRe	42.2	**78.2**	63.0
$\text{xCoRe}_{\text{ECB+}}$	**42.5**	78.0	61.9

Table 4: Long-document comparison systems scores (CoNLL-F1) when trained on LitBank and tested on LitBank and Animal Farm, and when trained and tested on BookCoref. (-) indicates runs that cause out of memory.

in which, at test time, documents are split into multiple segments to simulate long-document constraints, and on *ECB+ Sampled*, in which only a subset of documents per topic is used. We note that, to ensure robust results on ECB+, when testing with a subset of n documents, we average the results of 10 different runs in which each topic of the ECB+ test set has only n randomly selected documents.

Interestingly, cluster merging obtains the best performance over the two alternative clustering strategies. Furthermore, we observe that the performance gap widens as the number of contexts increases, highlighting the reliability of our technique when multiple contexts are provided. Moreover, our cross-context merging strategy convincingly outperforms the traditional mention-to-antecedent approach, confirming the superiority of our method based on merging locally extracted clusters.

5.2 Cross-document Benchmarks

In Table 3 we report cross-document results on ECB+ and SciCo, showing that xCoRe improves significantly over PMCoref, the previous state-of-the-art technique for cross-document coreference resolution with predicted mentions. More interestingly, we report additional performance gains when pretraining our model on LitBank: on ECB+, $\text{xCoRe}_{\text{LitBank}}$ reaches 42.4 CoNLL-F1 points, +8.7 points over the previous best scores of PMCoref, and +2.1 points over its non-pretrained version. Similarly, on SciCo, our pretrained model records a best score of 30.5 CoNLL-F1, surpassing the previous state of the art by +7.2 points and our version with no additional pretraining by +2.7 points. This highlights one of the key advantages of our cross-context formulation, which is that it allows models to benefit from additional shared training data, something that was unexplored by past cross-document solutions. We also report that CDLM is still the best technique when starting from gold mentions. Nevertheless, this solution is not applicable in real-world scenarios in which models start from raw texts, and has been criticized for its high time and memory costs (Hsu and Horwood, 2022).

5.3 Long-document Benchmarks

As outlined in Table 4, xCoRe achieves robust performance on every long-document benchmark. On the Animal Farm benchmark, xCoRe surpasses all comparison systems, achieving a +5.9 CoNLL-F1 improvement over Dual-cache, the previous leading system. On LitBank, xCoRe reports a CoNLL-F1 score of 78.2, aligning closely with Maverick, the current state-of-the-art model in this setting. On BookCoref, xCoRe achieves robust results, with slightly better performance compared to Maverick, a system that adopts the traditional one-pass mention-to-antecedent strategy. However, on this benchmark, xCoRe cannot perform at the level of Longdoc. After reviewing an array of qualitative outputs of these two models, we believe that this score discrepancy is due to the different errors that those models produce: while xCoRe outputs better within-window predictions, it occasionally wrongly splits long coreference chains, producing, on average, 45 chains per document on BookCoref; on the other hand, Longdoc sometimes wrongly merges different entity mentions into the same coreference cluster, obtaining, on average, only 14 chains per document. While it is hard for humans to evaluate whether one of those two errors is more important,

Model	*Cross-document*		*Long-document*				*Medium-size*	
	ECB+	SciCo	Animal Farm	LitBank		BookCoref	OntoNotes	PreCo
				full	*4-splits*			
xCoRe	40.3	27.8	42.2	78.2	77.1	62.9	83.2	87.1
xCoRe$_{\text{gold mentions}}$	73.8	62.3	58.9	88.2	85.6	64.0	89.2	94.8
xCoRe$_{\text{gold mentions \& gold clusters}}$	77.4	68.8	62.7	100.0	92.3	78.4	100.0	100.0

Table 5: Step-wise error analysis of xCoRe performance using gold information on all tested datasets in terms of CoNLL-F1 score. In particular, we detail the results of xCoRe with a version that starts from gold mentions (performing clustering and merging steps) and a version that starts from gold clusters (performing merging only).

empirical results show that the former error has a greater negative effect on the overall CoNLL-F1 score, as also demonstrated by several previous works (Moosavi and Strube, 2016; Duron-Tejedor et al., 2023; Martinelli et al., 2025).

We also note that, differently from what we saw for the cross-document scenario, in this case, pretraining on additional cross-document data does not yield meaningful gains. This outcome is likely due to the higher quality of LitBank annotations, which provide more stable training feedback compared to the noisier supervision often found in cross-document datasets. Finally, we highlight that Hierarchical (Gupta et al., 2024) particularly underperforms in the long-document scenario due to its limitations of filtering out singleton mentions from each small context, a problem that inevitably accumulates with very long documents.

5.4 Medium-size Benchmarks

In Table 5, we can find the results of xCoRe on the OntoNotes and PreCo medium-size benchmarks (first row). We report scores that are in the same ballpark as the current state-of-the-art system, Maverick (Martinelli et al., 2024), which was also used as the xCoRe underlying technique for within-context coreference resolution. While on one hand, this result is inherently implied by our pipeline design, on the other hand, it further demonstrates the generalization capabilities of our training strategy.

5.5 Step-wise Error Analysis

To further analyze the effectiveness of our pipeline, in Table 5 we report the performance of xCoRe on all of our tested datasets, along with an oracle-style step-wise analysis over each step of the xCoRe pipeline. Specifically, we compare our model performance against two baselines, in which i) we start from gold mentions, skipping the mention clustering step, and ii) we use both gold mentions and gold clusters, therefore only executing the cluster merging approach.

We report that, across datasets, with the exception of BookCoref, adopting an oracle mention extraction step by using gold mentions is especially beneficial. Indeed, a notable decrease in errors is shown in the cross-document setting, which suggests that mention identification is the main bottleneck when dealing with mentions across documents. This is not true on the BookCoref benchmark, because they only annotate book characters and therefore mention identification is easier. Furthermore, we can observe that using an oracle mention clustering step does not bring substantial benefit to our automatic pipeline when dealing with cross-context scenarios: in this case, the bottleneck is cluster merging. This result suggests that focusing on advancing our proposed simple yet effective cluster merging technique could lead to additional improvements in every coreference scenario.

6 Conclusion

In this paper, we introduce the cross-context coreference resolution setting, a generalization of classical coreference that includes medium-size , long- and cross-document settings. We also propose xCoRe, an all-in-one coreference resolution system that uses a three-step pipeline to extract mentions and clusters locally, and then merge them across contexts. In our experiments, we show that framing coreference as a cross-context problem enables training on shared resources, thereby making it possible to use additional data to improve model performance. More importantly, we demonstrate that our new architecture attains new state-of-the-art scores on cross-document benchmarks and top-tier results on both medium-size and long-document datasets. We believe that, by releasing this model, we could potentially benefit several downstream applications, filling the gap for an end-to-end, robust system across challenging coreference scenarios.

7 Limitations

Our experiments are limited to English entity coreference resolution, and we do not explore xCoRe capabilities in other languages or coreference settings, such as event coreference. However, our model is language-agnostic, and our technique can be naturally extended to events without the need for additional heuristics. We leave this as future work. Furthermore, all of our experiments were limited by our resource setting, i.e., a single RTX-4090. This has impacted our training and evaluation for long-document results, such as on BookCoref, in which our maximum window size for training xCoRe models was only 1500 tokens, and with the benchmarking of autoregressive models, such as seq2seq (Zhang et al., 2023), which require a more resourceful hardware setup. Nevertheless, we believe this limited setting is a common scenario in many real-world applications that would substantially benefit from adopting xCoRe as their all-in-one coreference system.

Acknowledgements

We gratefully acknowledge the support of the PNRR MUR project PE0000013-FAIR.

We also gratefully acknowledge the support of the AI Factory IT4LIA project. This work has been carried out while Giuliano Martinelli was enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome.

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.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

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

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2021a. Cross-document coreference resolution over predicted mentions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5100–5107, Online. Association for Computational Linguistics.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2021b. Realistic evaluation principles for cross-document coreference resolution. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 143–151, Online. Association for Computational Linguistics.

Arie Cattan, Sophie Johnson, Daniel Weld, Ido Dagan, Iz Beltagy, Doug Downey, and Tom Hope. 2021c. Scico: Hierarchical cross-document coreference for scientific concepts. *Preprint*, arXiv:2104.08809.

Hong Chen, Zhenhua Fan, Hao Lu, Alan Yuille, and Shu Rong. 2018. PreCo: A large-scale dataset in preschool vocabulary for coreference resolution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 172–181, Brussels, Belgium. Association for Computational Linguistics.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4545–4552, Reykjavik, Iceland. European Language Resources Association (ELRA).

Ana-Isabel Duron-Tejedor, Pascal Amsili, and Thierry Poibeau. 2023. How to Evaluate Coreference in Literary Texts? *Preprint*, arXiv:2401.00238.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Qipeng Guo, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2023. Dual cache for long document neural coreference resolution. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15272–15285, Toronto, Canada. Association for Computational Linguistics.

Talika Gupta, Hans Ole Hatzel, and Chris Biemann. 2024. Coreference in long documents using hierarchical entity merging. In *Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024)*, pages 11–17, St. Julians, Malta. Association for Computational Linguistics.

Benjamin Hsu and Graham Horwood. 2022. Contrastive representation learning for cross-document coreference resolution of events and entities. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3644–3655, Seattle, United States. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Lauri Karttunen. 1969. Discourse referents. In *International Conference on Computational Linguistics COLING 1969: Preprint No. 70*, Sånga Säby, Sweden.

Nghia T. Le and Alan Ritter. 2023. Are large language models robust coreference resolvers? *Preprint*, arXiv:2305.14489.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Gili Lior, Avi Caciularu, Arie Cattan, Shahar Levy, Ori Shapira, and Gabriel Stanovsky. 2024. Seam: A stochastic benchmark for multi-document tasks. *Preprint*, arXiv:2406.16086.

Yanming Liu, Xinyue Peng, Jiannan Cao, Shi Bo, Yanxin Shen, Tianyu Du, Sheng Cheng, Xun Wang, Jianwei Yin, and Xuhong Zhang. 2025. Bridging context gaps: Leveraging coreference resolution for long contextual understanding. *Preprint*, arXiv:2410.01671.

Giuliano Martinelli, Edoardo Barba, and Roberto Navigli. 2024. Maverick: Efficient and accurate coreference resolution defying recent trends. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13380–13394, Bangkok, Thailand. Association for Computational Linguistics.

Giuliano Martinelli, Tommaso Bonomo, Pere-Lluís Huguet Cabot, and Roberto Navigli. 2025. BOOKCOREF: Coreference resolution at book scale. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24526–24544, Vienna, Austria. Association for Computational Linguistics.

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.

Shon Otmazgin, Arie Cattan, and Yoav Goldberg. 2023. LingMess: Linguistically informed multi expert scorers for coreference resolution. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2752–2760, Dubrovnik, Croatia. Association for Computational Linguistics.

Ian Porada, Xiyuan Zou, and Jackie Chi Kit Cheung. 2024. A controlled reevaluation of coreference resolution models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 256–263, Torino, Italia. ELRA and ICCL.

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.

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.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 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.

Lixing Zhu, Jun Wang, and Yulan He. 2025. Llm-Link: Dual LLMs for dynamic entity linking on long narratives with collaborative memorisation and prompt optimisation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11334–11347, Abu Dhabi, UAE. Association for Computational Linguistics.

A Additional Details on within-Context Coreference

The within-context component of the xCoRe architecture is responsible for extracting mentions and clustering them locally. To do this, we adopt the mention extraction pipeline presented in Maverick (Martinelli et al., 2024) and the mention clustering strategy adopted in LingMess (Otmazgin et al., 2023), proven to be an optimal combination in previous works.

We now report the details of our two within-context coreference resolution steps, namely, mention extraction and mention clustering.

A.1 Mention Extraction

For any input context, mention spans are extracted within a single context in two steps. First, the model predicts candidate start positions for mentions, and then, for each predicted start, it predicts potential end positions. Let $(x_1, \ldots, x_n)$ be the contextualized token embeddings of input context $c = (t_1, \ldots, t_n)$. The probability of token t_i being the start of a mention is computed as:

$$F_{\text{start}}(x) = W'_{\text{start}}(\text{GeLU}(W_{\text{start}}x))$$
$$p_{\text{start}}(t_i) = \sigma(F_{\text{start}}(x_i))$$

For each t_s such that $p_{\text{start}}(t_s) > 0.5$, the model then scores subsequent tokens t_j (with $s \leq j$) as potential mention ends, conditioned on the start token:

$$F_{\text{end}}(x_s, x_j) = W'_{\text{end}}(\text{GeLU}(W_{\text{end}}[x_s, x_j])),$$
$$p_{\text{end}}(t_j \mid t_s) = \sigma(F_{\text{end}}(x_s, x_j))$$

The model considers only tokens up to the next sentence boundary. This strategy, called end-of-sentence (EOS) mention regularization, significantly narrows the span search space, reducing computational cost without sacrificing recall.

A.2 Mention Clustering

Once mentions have been extracted from an individual context, we score coreference links between mention pairs using a multi-expert architecture that assigns a specialized scorer to each pair based on its linguistic type. We follow the classification proposed by Otmazgin et al. (2023), which partitions mention pairs into six categories, as reported below:

- PRON-PRON-C: Compatible pronouns (e.g., *(I, I), (he, him)*)
- PRON-PRON-NC: Incompatible pronouns (e.g., *(I, he)*)
- ENT-PRON: Pronoun and non-pronoun (e.g., *(Mark, he)*)
- MATCH: Identical content (e.g., *(New York, New York)*)
- CONTAINS: Nested or partial match (e.g., *(Barack Obama, Obama)*)
- OTHER: All remaining cases

Each category k_g has a dedicated mention-pair scorer. Given a mention $m_i = (x_s, x_e)$ and a candidate antecedent $m_j = (x_{s'}, x_{e'})$, each mention boundary is encoded with a category-specific linear layer:

$$F_s^{k_g}(x) = W'_{k_{g,s}}(\text{GeLU}(W_{k_{g,s}}x))$$
$$F_e^{k_g}(x) = W'_{k_{g,e}}(\text{GeLU}(W_{k_{g,e}}x))$$

The final coreference score $p_c^{k_g}(m_i, m_j)$ is computed using a bilinear interaction between all combinations of start and end embeddings:

$$\begin{aligned} p_c^{k_g}(m_i, m_j) = \sigma(&F_s^{k_g}(x_s) \cdot W_{ss} \cdot F_s^{k_g}(x_{s'}) + \\ &F_e^{k_g}(x_e) \cdot W_{ee} \cdot F_e^{k_g}(x_{e'}) + \\ &F_s^{k_g}(x_s) \cdot W_{se} \cdot F_e^{k_g}(x_{e'}) + \\ &F_e^{k_g}(x_e) \cdot W_{es} \cdot F_s^{k_g}(x_{s'})) \end{aligned}$$

Here, $W_{ss}, W_{ee}, W_{se}, W_{es}$ are shared across categories, while the feedforward weights are specific to each type.

B Additional Loss Details

B.1 Training

The xCoRe architecture is trained end-to-end with a multitask objective that mirrors the three stages of our pipeline: within-context mention extraction, within-context mention clustering, and cross-context cluster merging using Binary Cross Entropy (BCE) loss:

$$L_{\text{coref}} = L_{\text{extr}} + L_{\text{clust}} + L_{\text{merge}}$$

Binary cross-entropy We define the binary cross-entropy loss as:

$$\ell_{\text{BCE}}(y,\, p) = -y \log(p) - (1-y)\log(1-p)$$

Mention extraction loss The mention extraction step is trained with a loss that supervises both the prediction of mention starts and the identification of their corresponding ends, as detailed in Section A.1. Therefore, given all contexts $c_i \in B$, where B is the training batch prepared with our training strategy detailed in Section A.2, i.e., we compute the mention extraction loss L_{extr} as:

$$L_{\text{start}}(c_i) = \sum_{j=1}^{N} \ell_{\text{BCE}}(y_j,\, p_{\text{start}}(t_j))$$

$$L_{\text{end}}(c_i) = \sum_{s=1}^{S} \sum_{k=1}^{E_s} \ell_{\text{BCE}}(y_{sk},\, p_{\text{end}}(t_k|t_s))$$

$$L_{\text{extr}} = \sum_{i=1}^{|B|} L_{\text{start}}(c_i) + L_{\text{end}}(c_i)$$

Here, N is the number of input tokens in the context, S is the number of predicted start positions, and E_s is the number of candidate end tokens considered for a given start t_s. The label y_j indicates whether token t_j begins a mention, and y_{sk} indicates whether token t_k completes a mention that begins at t_s. Our loss is the sum of the extraction losses for each context $c_i \in B$

Mention clustering loss To train the mention-level clustering component, we apply Binary Cross Entropy loss (BCE) over all mention pairs. For every mention m_k inside a given context c_i in the training batch B, the model considers all preceding mentions $m_k \in c_i$ as potential antecedents, and predicts whether they belong to the same coreference cluster. The loss is computed as:

$$L_{\text{clust}}(c_i) = \sum_{j=1}^{|M|} \sum_{k=1}^{|M|} \ell_{\text{BCE}}(y_{jk},\, p_c(m_j|m_k))$$

$$L_{\text{clust}} = \sum_{i=1}^{|B|} L_{\text{clust}}(c_i)$$

Here, $|M|$ is the number of predicted mentions in the current context, $y_{jk} \in \{0, 1\}$ indicates whether m_j and m_k refer to the same entity, and $p_c(m_j|m_k)$ is the model's predicted coreference score for the pair, computed using the category-specific mention-pair scorers described in Appendix A.2.

Cross-context cluster merging loss. We supervise the final stage of the pipeline by comparing clusters across different contexts c_i of the training batch B. We use CB to indicate the number of clusters extracted in the previous clustering step, $CB = |\{\mathcal{W}^{c_i}\}_{c_i \in B}|$, and define the cluster merging loss as :

$$L_{\text{merge}} = \sum_{a=1}^{CB} \sum_{\substack{b=1 \\ b \neq a}}^{CB} \ell_{\text{BCE}}(y^i_{ab},\, p_{\text{cm}}(\mathcal{W}^{c_i}_a, \mathcal{W}^{c_j}_b))$$

where $\mathcal{W}^{c_i}_a$ and $\mathcal{W}^{c_j}_b$ are clusters from local contexts $\{\mathcal{W}^{c_i}\}_{c_i \in B}$ and p_{cm} is defined in Equation 3.2.2. We do not calculate the loss for clusters that come from the same context, i.e., $c_i = c_j$, since they have already been predicted separately by the cluster merging step. This loss guides the final step of the pipeline by training the model to correctly predict whether two clusters from separate contexts $\mathcal{W}^{c_i}_a$ and $\mathcal{W}^{c_j}_b$ refer to the same entity.

Training details All models are trained end-to-end using supervised fine-tuning. Specifically, we use teacher forcing and calculate loss for each step on gold information. For mention extraction, end predictions are conditioned on gold start positions. For clustering and merging, losses are computed using gold mentions and gold clusters to isolate each stage of the pipeline.

C Additional Training Details

C.1 Datasets

Cross document datasets We note that for both our settings, we use the non-singleton, entity-only version of the dataset.

- **ECB+** is a well-established dataset used in cross-document coreference resolution based on news stories. ECB+ organizes documents in topics, and coreference relations cannot be found across different topics. It includes annotations for both within-document and cross-document coreference, and for both event coreference resolution and entity coreference resolution, considering entities only when participating in an event. A small portion of each document, handpicked and manually curated, known as the "Cybulska setting", is used for model evaluation. Although annotated predictions are limited to this subset, previous systems, such as PMCoref, have access to the context of the whole document. This approach is what we refer to as "additional context" in this paper. In our evaluation, we only test models without access to additional information, to uniform evaluation strategies, and to obtain a more straightforward and realistic setting.

- **SciCo** is a dataset designed for evaluating coreference resolution across scientific documents. It focuses on linking mentions of scientific concepts (such as tasks, methods, and datasets) that appear in different papers. As one of the few available resources for cross-document coreference, SciCo plays a key role in our evaluation.
  Annotations in SciCo are obtained in a two-step fashion with a semi-automatic approach, following guidelines from previous work on data collection (Cybulska and Vossen, 2014). The process relies on automatically extracting likely coreferent mentions from a large corpus of papers. Annotators are then asked to build clusters and hierarchical relationships between mentions.

Long document datasets

- **LitBank** contains 100 works of fiction, in which every document has an average length of 2,000 tokens.
  Differently from previous coreference datasets, it contains an average document length that is four times longer than other traditional benchmarks such as OntoNotes. It is available in 10 different cross-validation folds and we perform our experiments on its first fold, LB_0. We evaluate our models using singleton mentions and report comparison systems' results on the same splits.

- **Animal Farm** is a long document benchmark consisting of George Orwell's novel, manually annotated for coreference resolution by Guo et al. (2023), with approximately 35,000 tokens, annotations over 20 characters, and 1,722 mentions.

- **BookCoref** is a book-scale coreference resolution benchmark consisting of 50 fully automatically annotated books, used for training and validation, and 3 manually annotated narrative texts.

Traditional Medium-size Datasets

- **OntoNotes** is a richly annotated corpus designed to support a wide range of natural language understanding tasks, including coreference resolution. It encompasses 3493 documents from multiple genres such as news articles, telephone conversations, weblogs, and talk shows, reaching more than 190,000 mentions and 1.6 million tokens.

- **PreCo** is an English dataset for coreference resolution. It contains 38k documents and 12.5M words, mostly from preschoolers' vocabulary. The authors have not released their official test set. To evaluate our models consistently with previous approaches, we use the official 'dev' split as our test set and retain the last 500 training examples for model validation.

C.2 Comparison System Details

As discussed in Section 4.2, we compare xCoRe against state-of-the-art models across standard-, cross-, and long-document coreference benchmarks.
Many results were taken directly from prior work; however, some of them had to be implemented to enable a proper comparison or to test them on new benchmarks. For PMCoref†, we report new results under comparable conditions. In particular, the original implementation predicts mentions within a curated subset of each document (the "Cybulska setting") while encoding the full document for scoring. To fairly compare with xCoRe , we repeated PMCoref's experiments, removing access to the additional context, resulting in lower performance. We also evaluated PMCoref† on SciCo to provide a predicted-mention baseline for that dataset.

For the long-document setting (results in Section 5.3), since the authors do not include the weights in the original repository, we adopt a recent implementation of the Hierarchical model[2].

C.3 Setup

In our experiments, xCoRe systems adopt DeBERTA−v3 *large*[3] as an encoder, which is downloaded from the Huggingface Transformers library (Wolf et al., 2020). We adopt this encoder because it has been shown to be effective on the coreference resolution task by previous works (Martinelli et al., 2024). All our experiments are run on an academic budget i.e., a single NVIDIA RTX-4090.

[2] https://github.com/CompNet/Tibert

[3] https://huggingface.co/microsoft/deberta-v3-large