SAPIENZA
UNIVERSITÀ DI ROMA

Extending Coreference Resolution to Long Texts: From Paragraphs to Full Books and Beyond

Dipartimento di Ingegneria Informatica, Automatica e Gestionale
Dottorato Nazionale in Intelligenza Artificiale (XXXVIII cycle)

Giuliano Martinelli
ID number 1915652

Advisor
Prof. Roberto Navigli

Academic Year 2024/2025

Thesis defended on 28/01/2026
in front of a Board of Examiners composed by:

Prof. Fosca Giannotti (chairman)

Prof. Alessandro Cimatti

Prof. Cristian Randieri

This thesis was reviewed by two external reviewers:

Prof. Junyi Jessy Li, *The University of Texas at Austin*

Prof. Jackie Chi Kit Cheung, *McGill University*

Extending Coreference Resolution to Long Texts: From Paragraphs to Full Books and Beyond

Sapienza University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: giuliano.martinelli97@gmail.com

Dedicato a

Tutti quelli che mi sono stati vicino

~ This page was intentionally left blank ~

Abstract

In recent years, Large Language Models have reshaped the landscape of Natural Language Processing, achieving remarkable performance across a wide range of tasks. However, while their performance is impressive on short texts, their capabilities remain limited when dealing with longer contexts: memory and computational costs grow rapidly, coherence degrades, and outputs can lack robustness or factual grounding. Coreference Resolution, a long-standing task that involves determining when expressions refer to the same entity, directly addresses these weaknesses. It remains a fundamental component of discourse understanding, reasoning, and information extraction, especially in applications involving lengthy documents such as articles, reports, or books. Despite steady progress, most neural approaches remain optimized for current benchmarks that mainly contain short inputs, making them ill-suited to the challenges of real-world deployment.

This thesis aims to enhance Coreference Resolution techniques in the era of Large Language Models, pushing the boundaries of efficiency, robustness, and scalability of neural-based methods, particularly when dealing with long texts.

We begin by introducing a novel encoder-only neural architecture that achieves state-of-the-art performance across a broad range of Coreference Resolution benchmarks. This model challenges the prevailing reliance on large generative architectures with high computational overhead, instead establishing an optimal balance between performance and efficiency. Next, motivated by the absence of resources for evaluating Coreference capabilities on extended contexts, we introduce a new long-document benchmark of fully annotated narrative books. This dataset extends the length limits of existing corpora by several orders of magnitude, revealing the persistent shortcomings of current neural models in handling very long texts. Building on these insights, we propose the first unified architecture capable of addressing long- and cross-document Coreference within a single framework. Joint modeling of these

two challenging scenarios enables shared learning and leads to new state-of-the-art results across multiple datasets. Finally, to enhance interpretability in Coreference evaluation, we propose a new approach that integrates semantic information into standard practices. Together, these contributions, along with the produced artifacts, advance Coreference Resolution toward robust, efficient, and interpretable systems suited for deployment in realistic, large-scale NLP applications.

~ This page was intentionally left blank ~

Contents

Chapter 1

Introduction

Coreference resolution (CR) has long been recognized as a fundamental challenge in Natural Language Processing (NLP). At its core, the task requires determining which expressions in text refer to the same underlying entity.
For instance, in the passage below:

Barack Obama visited **Rome**. **The president** has loved **the city**.

A human reader immediately infers that "the president" refers to Barack Obama and "the city" refers to Rome. However, for decades, NLP researchers have tackled the challenge of automatically resolving this task, approaching CR through a variety of rule-based, statistical, and, more recently, neural methods. The transition to neural architectures, particularly those leveraging pretrained language models, has brought remarkable gains in benchmark performance. However, these advances have come at the cost of efficiency, since neural CR systems often rely on large annotated datasets, demand substantial memory and compute resources, and exhibit limited scalability when applied to long texts.

In fact, unlike sentence-level NLP tasks such as Named Entity Recognition or Part of Speech tagging, Coreference Resolution is usually performed as a document-level task, which implies reasoning over long contexts. References to the same entity may be distributed across an entire document, or even span multiple documents, requiring models to reason over long contexts. Current systems, optimized primarily for short inputs, significantly degrade in performance and efficiency when

confronted with real-world texts, such as scientific articles, news reports, or narrative books. Furthermore, many of those state-of-the-art systems suffer from architectural limitations that make it infeasible to apply them to such long or multi-document settings.

This thesis addresses these challenges by introducing new methods, resources, and analyses aimed at making CR more robust and efficient, especially when dealing with large inputs, such as in long- or multi-document settings. Our contributions focus specifically on reducing memory consumption, improving scalability, and enhancing usability and evaluation to move Coreference Resolution closer to practical deployment in real-world applications involving long and diverse texts.

1.1 Motivation and Research Objectives

Understanding *who* or *what* a text refers to is one of the most fundamental aspects of Natural Language Comprehension. Coreference Resolution embodies this challenge by connecting surface-level expressions to the underlying entities and events they denote, linking linguistic form to discourse semantics. Coreference serves as an enabling technology for a wide range of downstream tasks, including Document Summarization (Falke et al., 2017; Pasunuru et al., 2021; Liu et al., 2021), Information Retrieval (Liu et al., 2024; Xu et al., 2025), Machine Translation (Stojanovski and Fraser, 2018; Ohtani et al., 2019; Yehudai et al., 2023), Dialogue Systems (Jang et al., 2025), Question Answering (Chai et al., 2022; Liu et al., 2024), and Large-scale Knowledge Base population (Li et al., 2020; Zaporojets et al., 2022), among others.

In each of these domains, failing to capture coreferences of underlying entities can negatively impact the outcome of these downstream applications, resulting in fragmented or redundant entity representations, incoherent summaries, erroneous fact aggregation, or contradictory dialogue states. It is therefore crucial to regard CR as an essential component in the broader pursuit of document understanding and discourse modeling.

Despite its importance, much of the historical progress in CR has occurred under relatively narrow and controlled conditions. Benchmark datasets such as OntoNotes

(Pradhan et al., 2012) have defined the state of the art for over a decade, primarily consisting of short to medium-length texts, including documents on the order of a few hundred tokens such as news articles, broadcast transcripts, and conversational dialogues. Within this setting, increasingly sophisticated models have demonstrated steady improvements, with reported scores suggesting that their performance is approaching that of human annotators.

However, this apparent success is deceptive, as current systems are heavily optimized on the few high-quality available resources, and struggle to be applied out-of-domain. This problem becomes particularly evident when applying those models to very long texts, such as scientific papers or narrative books, in which those systems exhibit significant limitations not only in performance but also in practical usability, as efficiency was not taken into account in their architectural design.

The transition to Transformer-based language models has amplified this tension, driving major progress but exposing new limitations. On the one hand, contextual Pretrained Transformer encoders such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and SpanBERT (Joshi et al., 2020) have provided richer representational capacity, capturing intricate syntactic and semantic regularities that earlier models could not. On the other hand, their reliance on full self-attention introduces severe computational constraints, as the memory and runtime complexity scale quadratically with sequence length. As a result, applying these models directly to long texts becomes computationally infeasible. Approximate strategies such as segmenting input into sliding windows or employing sparse attention mechanisms alleviate some of these computational burdens at the cost of fragmenting discourse, often leading to lower performance, as the same entity may be inconsistently represented across adjacent segments, undermining the very coherence that Coreference Resolution is intended to model.

The computational challenge is compounded by a set of equally pressing linguistic challenges. Coreference in extended discourse is rarely limited to simple pronoun resolution. In narrative books, entities usually evolve dynamically: a protagonist may first appear under a full name, later recur as a pronoun, a descriptive epithet, or even an implicit reference. In technical and scientific writing, entities may be

introduced through formal terminology, then abbreviated or replaced with acronyms across subsequent sections or even across related documents. In newswire and multi-document scenarios, entities may be referenced under shifting titles, roles, or temporal epithets. Moreover, these mentions are not distributed uniformly: in long documents, entities may disappear for thousands of tokens before reappearing, forcing systems to maintain representations over extremely long temporal gaps. Accurately modeling such phenomena requires approaches that integrate fine-grained local linguistic cues with global discourse-level structure, a capacity that current systems, optimized for short texts, usually lack.

Evaluation methodology further compounds the limitations of Coreference Resolution. Metrics such as MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and $CEAF_{\phi 4}$ (Luo, 2005) conceptualize CR as an intra-document clustering task, rewarding partition overlap but offering little insight into model errors. They have been criticized for failing to capture discourse-level phenomena, especially in long documents where entities reappear after extended gaps and chains contain thousands of mentions. In such settings, their scores are difficult to interpret or compare with respect to short-text performance.

Beyond this, unlike the majority of NLP tasks, metrics adopted for coreference evaluation are still based on statistical heuristics and are not neural or semantic-aware metrics for coreference that can leverage contextual understanding. This lack of semantic understanding can lead to serious inconsistencies, as those metrics require exact span matches and therefore semantically irrelevant boundary differences are treated as complete errors, complicating comparisons across annotations, genres, or languages. Not only that, interpretability is also a problem: reporting a single global score obscures systematic failure modes, for example, whether errors stem from missing rare entities or annotation conventions. Consequently, reported gains may reflect optimization for benchmark-specific conventions rather than genuine progress in discourse modeling.

These shortcomings highlight the need for evaluation methods that are capable of exposing semantically grounded error patterns, enabling more robust and interpretable assessments of CR systems.

This dissertation is motivated by these limitations. The central research question can be framed as follows:

> *How can we design Coreference Resolution systems that operate both efficiently and accurately in long- and cross-document settings, scaling from short paragraphs to entire books and multiple-document collections?*

To address this question, the thesis pursues four overarching objectives:

- Develop computationally scalable models that handle efficiently and accurately the Coreference Resolution task under realistic resource constraints.
- Extend Coreference to very long texts, filling the current gap in available resources to benchmark and train models in this scenario.
- Design an all-in-one framework, that integrate short-, long-, and cross-document CR within a single system.
- Advance interpretability by integrating semantic cues in coreference evaluation, showing meaningful error patterns and robustness under domain shift.

These objectives aim to move Coreference Resolution beyond short-text benchmarks toward methods that can operate at the scale, complexity, and diversity of real-world discourse.

1.2 Contributions

The contributions of this dissertation map directly to these objectives, addressing the barriers to scalable and reliable CR:

1. **Efficiency and Accuracy.** We introduce Maverick (Martinelli et al., 2024a), an efficient and accurate neural architecture that defies recent trends of using generative architectures for the Coreference Resolution task. We show that a simple yet efficient encoder-only pipeline can obtain the best performances on current traditional benchmarks, including OntoNotes, while allowing for processing under realistic computational constraints.

2. **Extending Coreference to Long Documents.** We develop and release BookCoref (Martinelli et al., 2025), the first annotated corpus of full-length books. BookCoref is composed of 50 high-quality automatically annotated books, which can serve as a reliable training set for long-document models. It also contains 3 manually annotated books, enabling, for the first time, a rigorous evaluation of CR systems at book length, revealing challenges that are invisible in short- and medium-length document benchmarks.

3. **Unifying Long- and Cross-document Coreference.** We propose *xCoRe*, the first unified model designed to address short-text, long-document, and cross-document CR within a single framework. By combining efficiency with high performance, we demonstrate that approaching long- and cross-document CR with a single architecture leads to further improvements.

4. **Semantically-enhanced Evaluation.** We incorporate semantic information into coreference scoring to expose systematic weaknesses hidden by traditional metrics. Through the creation and downstream application of the *CNER* task (Martinelli et al., 2024b), we demonstrate how semantic typing of mentions provides finer-grained insights into model behavior, error patterns, and domain transferability.

1.2.1 Thesis Outline

The structure of the thesis reflects this progression:

- **Chapter 2** surveys the background of Coreference Resolution, introducing the task, evaluation, datasets, current trends, and the challenges of long- and cross-document settings.

- **Chapter 3** presents *Maverick*, detailing its architecture, training pipeline, and efficiency advantages.

- **Chapter 4** introduces *BookCoref*, describing the silver and gold annotation pipelines, dataset statistics, and benchmark results on book-scale texts.

- **Chapter 5** develops *xCoRe*, our unified framework for handling coreference across short, long, and multi-document inputs.
- **Chapter 6** turns to semantic grounding, first introducing the *CNER* task and dataset and then adopting it to propose a complementary semantically-enhanced approach to improve interpretability in CR evaluation.
- **Chapter 7** concludes with a discussion of limitations and outlines future directions for advancing efficient and robust Coreference Resolution.

1.2.2 Publications

The research presented in this thesis has been jointly conducted at Sapienza NLP, the research group directed by Professor Roberto Navigli at Sapienza University of Rome, over the three years of the Ph.D. in Artificial Intelligence program. In what follows, we provide a reference to these publications in chronological order:

1. **Martinelli, G.**, Molfese, F., Tedeschi, S., Fernández-Castro, A., & Navigli, R. (NAACL 2024, June). CNER: Concept and Named Entity Recognition. *In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers) (pp. 8329-8344).*
2. **Martinelli, G.**, Barba E., & Navigli, R. (ACL 2024, June). 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) (pp. 13380–13394).*
3. **Martinelli, G.**, Bonomo T., Huguet-Cabot P., & Navigli, R. (ACL 2025, July). BOOKCOREF: Coreference Resolution at Book Scale. *In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 24526–24544).*
4. **Martinelli, G.**, Bruno G. & Navigli, R. (EMNLP 2025, November). xCoRe: Cross-context Coreference Resolution. *In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing .*

~ This page was intentionally left blank ~

Chapter 2

Background: Coreference Resolution

This chapter lays the groundwork for understanding the Coreference Resolution task. We begin with a formal definition and key terminology, followed by a discussion of three increasingly challenging CR settings: i) traditional short- and medium-length documents, ii) very long documents, and iii) multi-document collections. For each setting, we review the main challenges, available datasets, and modeling approaches. Finally, we reserve a more detailed introduction of evaluation methodologies, highlighting their current limitations.

2.1 The Task

Coreference Resolution is the task of identifying and grouping expressions in text that refer to the same underlying entity (Karttunen, 1969). These expressions are typically called *mentions* (or *markables*).
Mentions can take many forms, including, among others:

- **Proper nouns**, e.g., “Barack Obama” or “U.S.A.”.
- **Common nouns**, e.g., “the president” or “the country”.
- **Noun phrases**, e.g., “the tall man” or “the research team chief”.
- **Personal pronouns**, e.g., “he” or “she”.

- **Possessive pronouns**, e.g., "his" or "their".

Mentions that refer to the same entity are grouped into a *coreference cluster*, representing a distinct discourse entity, while mentions appearing only once are referred to as *singletons*. Furthermore, an additional challenge in determining mention relations and boundaries arises from the presence of *overlapping mentions*: e.g., in "Mark's father," the mention *Mark* refers to one entity, while *Mark's father* refers to another, related entity.

To illustrate the variety of coreference phenomena, consider the following examples:

1. **Pronoun Resolution**:

   [The cat]$_{\text{cat}}$ played with [its]$_{\text{cat}}$ toy

   Here, "the cat" and "its" refer to the same entity and, therefore, will be put in the same coreference cluster.

2. **Noun-phrases and Overlapping mentions**:

   [[Barack]$_{\text{Barack Obama}}$ and Michelle]$_{\text{Barack and Michelle Obama}}$ traditionally share [their]$_{\text{Barack and Michelle Obama}}$ [birthday cake]$_{\text{cake}}$.
   However, this time, [the president]$_{\text{Barack Obama}}$ decided to have [one]$_{\text{cake}}$ for [himself]$_{\text{Barack Obama}}$.

   This example illustrates several important cases: i) overlapping mentions ("Barack" within "Barack and Michelle"), ii) nominal variation ("Barack Obama" vs. "the president"), and iii) multiple mentions of the same non-person entity ("birthday cake" vs. "one").

3. **Ambiguity and World Knowledge**:

   [The sword]$_{\text{sword}}$ could not fit into [the box]$_{\text{box}}$ because [it]$_{\text{box}}$ was too small.

   Resolving "it" requires reasoning beyond syntax: depending on context, "it" could refer to either the sword or the box. Disambiguating these coreference

cases often requires commonsense or pragmatic cues (Govindarajan et al., 2024).

2.1.1 Formal Definition

We now present a formal definition of the above-introduced Coreference Resolution task.

Let a document D be represented as a sequence of tokens as:

$$D = (t_1, t_2, \ldots, t_n).$$

A mention m is a contiguous span of tokens in D that refer to a single entity, such as:

$$m = (t_s, \ldots, t_e), \quad 1 \leq s \leq e \leq n.$$

The set of all mentions in D is denoted by $M = \{m_1, \ldots, m_{|M|}\}$, and a goal of Coreference Resolution is to partition M into clusters as follows:

$$C = \{C_1, \ldots, C_k\}, \quad \bigcup_{i=1}^{k} C_i = M, \quad C_i \cap C_j = \emptyset \text{ for } i \neq j,$$

such that each cluster $C_i = \{m_{i_1}, \ldots, m_{iz}\}, z = |C_i|$ corresponds to a single entity. Equivalently, this define an coreference relation $\sim$ over mentions:

$$m_a \sim m_b \quad \Leftrightarrow \quad m_a \text{ and } m_b \text{ refer to the same entity.}$$

This formulation naturally extends to multi-document settings, where the token sequence $(t_1, \ldots, t_n)$ represents the concatenation of all tokens across a collection of documents.

In what follows, we first examine traditional Coreference Resolution techniques and resources operating on sentences, paragraphs, and medium-sized documents. We then introduce two scenarios that are central to this thesis and considerably more challenging: Coreference in very long documents and Cross-document Coreference.

2.2 Coreference Resolution on Short Documents

Over the past two decades, research on Coreference Resolution has predominantly focused on short, single-document texts, such as news articles and conversational transcripts. This emphasis arises from two main factors: first, the relative ease of modeling and evaluating Coreference systems within short texts compared to longer or multi-document settings; and second, the availability of high-quality annotated short-document corpora.

In the sections that follow, we first provide an overview of the primary resources developed for Coreference Resolution in short-document scenarios. We then survey the current methodologies that have proven most effective, highlighting both discriminative and generative approaches.

2.2.1 Datasets

We generally refer to *short- or medium-size document resources* as the datasets whose texts, on average, do not exceed approximately 1,500 tokens. Among these, **OntoNotes** (Pradhan et al., 2012) is by far the most influential manually annotated corpus. Serving as the foundation for the CoNLL-2011 and CoNLL-2012 shared tasks, it comprises 3,493 documents and covers over 190,000 mentions and 1.6 million tokens, spanning diverse genres including news articles, telephone conversations, and weblogs. The documents in OntoNotes are generally short, averaging fewer than 500 tokens, and the annotations focus on core nominal and pronominal mentions, excluding singletons. This design choice simplified evaluation and facilitated a clear benchmark for clustering quality, but it also narrowed the scope of the task: systems trained on OntoNotes often overfit to genre-specific patterns and struggle to generalize to longer documents or benchmarks that include singleton mentions. Despite these limitations, OntoNotes remains the de facto standard benchmark and continues to influence model architectures, largely because it provides a high-quality, multi-genre training set.

PreCo (Chen et al., 2018) offers a larger-scale dataset with respect to the limited scale of OntoNotes. It contains 38,000 documents and 12.5 million words, primarily drawn from vocabulary suited for preschool reading levels. Unlike OntoNotes,

PreCo includes singleton mentions—constituting more than half of all mentions in the corpus—leading to a significantly higher mention density per token. This richer annotation scheme exposes models to more challenging decisions regarding entity boundaries and linking. However, documents in PreCo are even shorter on average than those in OntoNotes, and therefore entities never recur after long gaps, with a discourse structure that remains relatively shallow compared to longer texts.

Other benchmarks were designed to probe specific weaknesses of models trained on OntoNotes and PreCo. The **GAP** dataset (Webster et al., 2018) focuses on gender-ambiguous pronoun resolution, with each sample presenting two sentences in which a model must identify the correct antecedent for a candidate mention among two antecedents with different genders. This task highlights both gender biases and limitations in handling ambiguous references, and GAP has become a standard diagnostic tool for evaluating fairness in Coreference Resolution and pretrained Transformer models. **WikiCoref** (Ghaddar and Langlais, 2016), in contrast, is a full-document corpus derived from Wikipedia. It was developed to assess models on out-of-domain text, as Wikipedia articles differ substantially from the news and conversational data that dominate OntoNotes. Models trained exclusively on OntoNotes often underperform on WikiCoref, revealing challenges in cross-domain generalization and sensitivity to annotation guidelines, which we revisit in the next chapter.

Annotation differences

A persistent challenge in Coreference Resolution benchmarks lies in the diversity of annotation conventions (Porada et al., 2024a), which affects model performance, generalization, and cross-dataset evaluation. Key sources of variation include:

- **Inclusion of Singletons:** OntoNotes excludes singleton mentions (i.e., expressions that refer to an entity without any co-referential mention), whereas other benchmarks, such as PreCo, include them. This distinction influences both model training and evaluation: systems trained on OntoNotes often fail to generalize well to datasets that include singletons, as they have not learned to recognize such mentions.

- **Mention Types:** Annotation guidelines often differ in what qualifies as a mention. Some corpora annotate a broad range of expressions, such as proper nouns, common nouns, noun phrases, pronouns, and descriptive roles, while others include only a subset of these categories. In addition, certain resources focus on a limited set of semantic classes (Bamman et al., 2020), such as *People*, *Facilities*, and *Locations*, inter alia, whereas others aim to annotate every entity mention that annotators can identify, regardless of its semantic type.

- **Coreference Relations:** Conventions differ in how they treat specific phenomena such as appositions, copular constructions, generic mentions, cataphora, and exophora. These choices can substantially change the characteristics and size of coreference chains. For instance, some resources explicitly annotate *zero pronouns*, i.e., omitted or implicit pronouns that are recoverable from context, while others ignore them altogether.

- **Overlapping Mentions:** Some schemes allow nested mentions (e.g., "*Barack*" inside "*Barack Obama*"), while others annotate only the largest span. Such differences alter the granularity of mention boundaries and cluster composition.

- **Span Length and Granularity:** The length and syntactic constituency of annotated mentions can vary widely. For example, in PreCo, "Secretary of State Colin Powell" is annotated as two distinct and coreferring mentions, i.e., "Secretary of State" and "Colin Powell", while in OntoNotes it would be tagged as a single continuous mention. This influences both the density and interpretability of clusters.

- **Domain and Genre Effects:** Annotation practices often adapt to text genre, usually focusing on frequent entity types, creating further inconsistency across corpora.

These discrepancies collectively shape the density, structure, and coverage of coreference chains, leading to variations in model performance and hindering cross-domain generalization. The impact of such annotation differences becomes even

more pronounced in long-document and cross-document settings, where coreference mentions span across larger contexts, and inconsistencies compound.

2.2.2 Current Approaches

Modern approaches to Coreference Resolution can be broadly divided into two families: discriminative models, which treat the task as classification over candidate spans, and generative models, which cast it as a sequence generation problem. Although they differ in formulation, both paradigms have been shaped by the availability of pretrained language models and benchmark datasets such as OntoNotes, and both face important challenges when applied beyond short texts, which we will present in Section 2.3.2.

Discriminative Models Discriminative models were the first to become the standard approach for coreference resolution. A major breakthrough came with the end-to-end formulation introduced by Lee et al. (2017, 2018), commonly known as the Coarse-to-Fine architecture. This technique operates in two stages: First, it involves a mention extraction step, in which the spans most likely to be coreferent mentions are identified. This is followed by a mention-antecedent classification step where, for each extracted mention, the model searches for its most probable antecedent (i.e., the extracted span that appears before in the text).

This pipeline, composed of mention extraction and mention-antecedent classification steps, has been adopted with minor modifications in many subsequent works, which we refer to as *Coarse-to-Fine* models. Among those works Lee et al. (2018), Joshi et al. (2019) and Joshi et al. (2020) experimented with changing the underlying document encoder, utilizing ELMo (Peters et al., 2018), BERT (Devlin et al., 2019) and SpanBERT (Joshi et al., 2020), respectively, achieving remarkable score improvements on the English OntoNotes. Similarly, Kirstain et al. (2021) introduced s2e-coref that reduces the high memory footprint of SpanBERT by leveraging the LongFormer (Beltagy et al., 2020a) sparse-attention mechanism. Based on the same architecture, Otmazgin et al. (2023) analyzed the impact of having multiple experts score different linguistically motivated categories (e.g., pronouns-nouns, nouns-nouns,

etc.).

An alternative discriminative paradigm is represented by models that resolve coreference resolution incrementally. Incremental Coreference Resolution has a strong cognitive grounding: research on the "garden-path" effect shows that humans resolve referring expressions incrementally (Altmann and Steedman, 1988a). A seminal work that proposed an automatic incremental system was that of Webster and Curran (2014), which introduced a clustering approach based on the shift-reduce paradigm. In this formulation, for each mention, a classifier decides whether to SHIFT it into a singleton (i.e., single mention cluster) or to REDUCE it within an existing cluster. The same approach has recently been reintroduced in ICoref (Xia et al., 2020) and longdoc (Toshniwal et al., 2021), which adopted SpanBERT and LongFormer, respectively.

Generative models Recent state-of-the-art coreference resolution systems, before our work, have predominantly adopted autoregressive generative approaches. The first system to show that the autoregressive formulation was competitive was ASP (Liu et al., 2022), which outperformed encoder-only discriminative approaches. ASP is an autoregressive pointer-based model that generates actions for mention extraction (bracket pairing) and then conditions the next step to generate coreference links. Notably, the breakthrough achieved by ASP is not only due to its formulation but also to its usage of large generative models. Indeed, the success of their approach is strictly correlated with the underlying model size, since, when using models with a comparable number of parameters, the performance is significantly lower than encoder-only approaches. The same occurs in Zhang et al. (2023), a fully-seq2seq approach where a model learns to generate a formatted sequence encoding coreference notation, in which they report a strong positive correlation between performance and model sizes.

Finally, the system that held state-of-the-art scores on the OntoNotes benchmark before our work was held by Link-Append (Bohnet et al., 2023), a transition-based system that incrementally builds clusters exploiting a multi-pass Sequence-to-Sequence architecture. This approach incrementally maps the mentions in previously

coreference-annotated sentences to system actions for the current sentence, using the same shift-reduce incremental paradigm presented before for incremental models. While the foregoing models ensured superior performance compared to previous discriminative approaches, using them for inference was out of reach for many users, not to mention the exorbitant cost of training them from scratch.

2.3 Coreference Resolution on Long Documents

As discussed in the previous section, most progress in Coreference Resolution has been driven by benchmarks such as OntoNotes, whose documents are relatively short, typically containing a few hundred tokens. While models trained in this regime are well-suited for short- to medium-sized texts, they often struggle when applied to long documents: they may be unable to process the entire input due to architectural constraints, or they incur prohibitive memory costs and suffer notable drops in accuracy.

Furthermore, the challenges of long-document Coreference are not only technical. Extended texts, such as novels, technical reports, and screenplays, exhibit richer discourse phenomena: entities often reappear after thousands of tokens—sometimes across chapters—and their surface forms evolve over time (e.g., a character introduced by full name is later referred to by a title, pronoun, or epithet). Dealing with this extensive length can make automatic models produce incoherent discourse (Xu et al., 2022) or forget crucial entities (Liu et al., 2024), and inherently makes annotation itself more labor-intensive: annotators must track many entities over long gaps, leading to higher costs and more inconsistent labels. As we will see in the next subsection, this complexity has resulted in a relative scarcity of high-quality long-document benchmarks.

2.3.1 Datasets

Most established English CR datasets were designed for short- or medium-length texts and often segment source documents to simplify annotation. For example, OntoNotes, the de facto standard for CR, has an average document length of

about 467 tokens. This brevity arises from splitting source documents into smaller partitions to simplify annotation. Shridhar et al. (2023) manually merge coreference clusters across partitions to construct full-document annotations,i.e., **LongtoNotes**, but, due to the short nature of the source genre, the average length only increases to 679 tokens. Beyond news and conversational text, which fall short of modeling long documents, a few datasets attempt to expose models to this scenario. The most well-known dataset is **LitBank** (Bamman et al., 2020), 100 excerpts from the literature genre, where documents are long, and relatively few major entities play a central role. However, while LitBank has become the long-document standard for CR, it truncates book samples to 2,000 tokens, failing to capture coreference relations that develop across entire books. **MovieCoref** (Baruah et al., 2021; Baruah and Narayanan, 2023) pushes document length further, with six full-length screenplays and three excerpts reaching about 20,000 tokens on average. However, the collection is small, and its screenplay-specific discourse structure limits generalization to other genres. A complementary resource is the manually annotated edition of Orwell's **Animal Farm** (Guo et al., 2023), which focuses on character-centric coreference in a single book.

2.3.2 Current Approaches

Among the short-document solutions already presented in Section 2.2.2, only a few apply to the long-document scenario.

Generative approaches, including Link-Append (Bohnet et al., 2023) and sequence-to-sequence models such as Zhang et al. (2023), treat CR as text generation, and are even more impractical when used in the long-document scenarios. Requiring the model to reproduce the input text along with its coreference annotations effectively doubles the input length and, combined with the fixed context window of large Transformers, makes processing entire books computationally prohibitive. The same concerns apply to Large Language Models (LLMs), whose application to CR is still under discussion, as current methods for LLM-based CR have yet to reach the performance of supervised models (Le and Ritter, 2024; Porada and Cheung, 2024; Gan et al., 2024, 2025).

Discriminative encoder-only models are generally more memory-efficient (Otmazgin et al., 2022) but suffer from a similar limitation: recent approaches such as LingMess (Otmazgin et al., 2023) or s2e-coref (Kirstain et al., 2021) are constrained to respect the maximum input length of their underlying Transformer encoder, i.e., LongFormer (Beltagy et al., 2020b).

However, among those models, **incremental** approaches have emerged as an ad-hoc formulation that works particularly well for long documents. Longdoc (Toshniwal et al., 2020, 2021), as presented in the previous chapter, incrementally builds clusters while maintaining a cache of the most recent entities, which makes it very efficient when dealing with long documents. Dual cache (Guo et al., 2023) extends this idea by adding a secondary cache that retains less-frequently used entities to improve long-term context retention. However, even if with these two models is possible to encode very long texts, on the fully annotated *Animal Farm* their performance still reaches only about 36% CoNLL-F1, far below typical scores on OntoNotes-style corpora, underscoring how challenging long-document CR remains.

2.4 Coreference Resolution across Multiple Documents

Several studies have investigated methods for addressing the Coreference Resolution task across multiple texts. This setting is commonly referred to as *cross-document Coreference Resolution*, and requires systems to identify mentions of the same entity that occur in different documents and to link them consistently. It introduces a distinct set of challenges compared to the single-document setting. A first problem for current systems, similar to that in long-document coreference, is the large volume of text: models must reason over multiple documents while maintaining a coherent representation of entities across them. Further complexity also arises from the heterogeneity of document sources, as entities may appear under different lexical forms, abbreviations, or naming conventions depending on the temporal and contextual variation or differences in authorial style.

In this section, we review approaches to cross-document Coreference Resolution, focusing specifically on *end-to-end entity-based* models. Therefore, we do not cover the identification and linking of events, or approaches that rely on gold mentions, as

these require additional resources and preprocessing steps that limit their applicability in real-world scenarios (Cattan et al., 2021a). Our focus aligns with standard practice in traditional and long-document coreference research, dealing with models that first predict and then link entity mentions.

2.4.1 Datasets

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, being approximately three times larger than ECB+ and including 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.

2.4.2 Current Approaches

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., 2020a) 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.

2.5 Evaluation

We have so far reviewed Coreference Resolution across different settings, including the models and datasets used. We now turn to the evaluation of the Coreference task itself. As we saw from the previous chapters, models must correctly *detect* mentions, *link* them, and finally *cluster* all coreferent mentions of the same entity. No single metric fully captures all of these aspects, which makes evaluation both nuanced and sometimes inconsistent across studies. The most widely adopted evaluation paradigm stems from the OntoNotes benchmark, which established the core rules for the CoNLL-2011 and CoNLL-2012 shared tasks. The resulting *CoNLL-F1* score, an average of three complementary metrics, remains the de facto standard, as we describe below.

2.5.1 CoNLL-2012 Coreference Resolution Metrics

For measuring mention detection, linking, and clustering quality, the CoNLL-2012 shared task (Pradhan et al., 2012) popularized reporting the unweighted average of three metrics: **MUC** (Vilain et al., 1995), $\mathbf{B^3}$ (Bagga and Baldwin, 1998), and $\mathbf{CEAF}_{\phi\mathbf{4}}$ (Luo, 2005).

We briefly give an overview of those metrics, while also discussing the shortcomings of each metric. In all of our metrics descriptions, K is the set of gold (key) clusters and R is the set of predicted (response) clusters.

- **MUC** (Vilain et al., 1995) is a *link-based metric*. It evaluates the minimal number of coreference links that must be added to or removed from the system output to make it identical to the gold clusters, i.e., to transform R into K. For a gold cluster $k_i \in K$, let $p_R(k_i)$ be the number of distinct response clusters in R that intersect with k_i.

  The recall for MUC is defined as:

  $$\text{MUC}_{\text{recall}} = \frac{\sum_{k_i \in K}(|k_i| - p_R(k_i))}{\sum_{k_i \in K}(|k_i| - 1)},$$

  and precision is defined symmetrically by swapping K and R.

MUC is known to be the least discriminative Coreference Resolution metric (Bagga and Baldwin, 1998; Luo, 2005) since it is only based on the minimum number of missing/extra links in the response compared to the key entities. For instance, MUC does not differentiate whether an extra link merges two singletons or the two most prominent entities of the text. However, the latter error does more damage than the first one.

Not only that, as we will also report when dealing with very long coreference chains in the long-document scenario, MUC overestimates outputs in which entities are over-merged (Moosavi and Strube, 2016; Duron-Tejedor et al., 2023).

- **B^3** (Bagga and Baldwin, 1998) is a *mention-based metric*, and addresses MUC's bias toward link counts by computing precision and recall at the *mention* level.

  For each mention m_i that appears in the set of gold mentions M, let $K(m_i)$ denote the gold cluster containing m_i, and $R(m_i)$ denote the response cluster containing m_i. The B^3 recall is defined as:

$$B^3{}_{\text{recall}} = \frac{1}{|M|} \sum_{m_i \in M} \frac{|R(m_i) \cap K(m_i)|}{|K(m_i)|},$$

  Similar to MUC, B precision is computed by switching the role of the key and response entities.

  B^3 rewards models that correctly group mentions into clusters and gives more weight to larger clusters. However, it also exhibits a well-known *mention identification bias*: if a mention appears in a response cluster, it is treated as a resolved mention regardless of whether its coreference links within that cluster are correct. This can lead to overly optimistic scores when mention detection is accurate.

- **$CEAF_{\phi 4}$** evaluates coreference resolution by optimally aligning predicted and gold clusters in a one-to-one fashion. The similarity between a gold cluster

$k_i \in K$ and a response cluster $r_j \in R$ is measured by the Dice coefficient:

$$\phi_4(k_i, r_j) = \frac{2\,|k_i \cap r_j|}{|k_i| + |r_j|}.$$

These similarity scores are used as weights in a bipartite graph between gold and response clusters. The optimal alignment $\mathcal{A}^*$ is obtained by solving a maximum bipartite matching problem:

$$\mathcal{A}^* = \arg \max_{\mathcal{A}\,\text{one-to-one}} \sum_{(k_i, r_j) \in \mathcal{A}} \phi_4(k_i, r_j).$$

Once the optimal alignment is found, precision and recall are computed as:

$$\text{Precision} = \frac{\sum_{(k_i,r_j)\in\mathcal{A}^*} \phi_4(k_i, r_j)}{\sum_{r_j \in R} \phi_4(r_j, r_j)}, \quad \text{Recall} = \frac{\sum_{(k_i,r_j)\in\mathcal{A}^*} \phi_4(k_i, r_j)}{\sum_{k_i \in K} \phi_4(k_i, k_i)}.$$

By enforcing a one-to-one alignment, CEAF_{ϕ_4} mitigates some of the problems observed in MUC and B^3, but also disregards correctly resolved mentions that belong to unaligned clusters, often resulting in lower performance estimates.

We also highlight that other metrics, such as **BLANC** (Luo et al., 2014) and **LEA** (Moosavi and Strube, 2016), were proposed to better handle singletons and cluster-size effects. However, they remain less widely adopted because the CoNLL-2012 average of MUC, B^3, and CEAF has become the de facto standard.

Unlike the majority of NLP tasks, there is still no neural or semantic-aware metric for coreference that can leverage contextual understanding. This lack of semantic understanding is a problem because, for example, all three metrics require exact span matches: minor but semantically irrelevant boundary differences are treated as complete errors, complicating comparisons across annotations, genres, or languages. Not only that, interpretability is also a serious challenge: reporting a single global score obscures systematic failure modes, for example, whether errors stem from missing rare entities, mis-linking pronouns, or fragmenting salient characters. Similar scores may therefore correspond to qualitatively different behaviors, making progress difficult to interpret.

Chapter 3

Improving Accuracy and Efficiency in Coreference Resolution

Abstract

The limitations identified in the previous chapter, such as computational inefficiency and a lack of practical deployability, are the motivation for our first contribution. While recent generative architectures have advanced performance on short-text benchmarks, they come at the cost of enormous computational overhead. In this context, the following chapter introduces Maverick, an encoder-only model designed to challenge this trend. By introducing Maverick, a carefully designed yet simple pipeline, we enable the running of a state-of-the-art Coreference Resolution system within the constraints of an academic budget, outperforming models with up to 13 billion parameters with as few as 500 million parameters. Maverick achieves state-of-the-art performance on the well-established OntoNotes benchmark, training with up to 0.006x the memory resources and obtaining a 170x faster inference compared to previous state-of-the-art systems. In the following chapter, we extensively validate the robustness of the Maverick framework with an array of diverse experiments, reporting improvements over prior systems in data-scarce, long-document, and out-of-domain settings.

Source: This chapter is based on the ACL 2024 paper: *Maverick: Accurate and Efficient Coreference Resolution Defying Recent Trends* (Martinelli et al., 2024a).
Code: `http://www.github.com/sapienzanlp/maverick-coref`

3.1 Overview

The first major contribution of this thesis focuses on proposing a simple and efficient pipeline for Coreference Resolution that combines high performance with striking efficiency. While proposing this model, we defy recent trends that either explore methods to obtain reasonable performance optimizing time and memory efficiency (Kirstain et al., 2021; Dobrovolskii, 2021; Otmazgin et al., 2022), or strive to improve benchmark scores regardless of the increased computational demand (Bohnet et al., 2023; Zhang et al., 2023).

As presented in Section 2.2.2, solutions that aim for efficiency usually rely on discriminative formulations, frequently employing the Coarse-to-fine mention-antecedent classification method proposed by Lee et al. (2017). This pipeline first involves a mention extraction step, in which the spans most likely to be coreference mentions are identified, and then a mention-antecedent classification step, where, for each extracted mention, the model searches for its most probable antecedent (i.e., the extracted span that appears before in the text). This two-step Coreference Resolution method has been adopted with minor modifications in many subsequent works, which we refer to as *Coarse-to-Fine models*.

On the other hand, performance-centered solutions are nowadays dominated by general-purpose large Sequence-to-Sequence models (Liu et al., 2022; Zhang et al., 2023). Those methods require to generate a formatted sequence encoding coreference notation, and their performance is strictly correlated with the underlying model size.

In this chapter, we argue that discriminative encoder-only approaches for Coreference Resolution have been prematurely discarded in the pursuit of achieving state-of-the-art performances and deserve further studies. By proposing Maverick, we strike an optimal balance between high performance and efficiency, a combination that was missing in previous systems. Our framework enables an encoder-only model to achieve top-tier performances while keeping the overall size of the models less than one-twentieth of the current state-of-the-art system, and training within academic resources. Moreover, when reducing the size of the underlying transformer encoder, Maverick performs in the same ballpark as encoder-only efficiency-driven solutions while improving speed and memory consumption. Finally, we propose a

novel incremental Coreference Resolution method that, integrated into the Maverick framework, results in a robust architecture for out-of-domain and data-scarce settings.

3.2 The Maverick Framework

We now present the Maverick framework: we propose replacing the preprocessing and training strategy of Coarse-to-Fine models with a novel pipeline that improves the training and inference efficiency of Coreference Resolution systems. Furthermore, with the Maverick Pipeline, we eliminate the dependency on long-standing manually-set hyperparameters that regulate memory usage.

Building on top of our pipeline, we propose three models that adopt a mention-antecedent classification technique, namely $\text{Maverick}_{\text{s2e}}$ and $\text{Maverick}_{\text{mes}}$, and a system that is based upon a novel incremental formulation, $\text{Maverick}_{\text{incr}}$.

3.2.1 Maverick Pipeline

The Maverick Pipeline is a combination of i) a novel mention extraction method, ii) an efficient mention regularization technique, and iii) a new mention pruning strategy.

Mention Extraction

When it comes to extracting mentions from a document D, there are different methods to model the probability that a text span contains an entity mention. Previous work follow the Coarse-to-Fine formulation, which consists of scoring all the possible spans in D. This entails a quadratic computational cost in relation to the input length, which they mitigate by introducing several pruning techniques.

With the Maverick Pipeline, we employ a different strategy. We extract coreference mentions by first identifying all the possible starts of a mention, and then, for each start, extracting its possible end. Specifically, to extract start indices, we first compute the hidden representation $(x_1, \ldots, x_n)$ of the tokens $(t_1, \ldots, t_n) \in D$ using a transformer encoder, and then use a fully-connected layer F to compute the

probability for each t_i being the start of a mention as:

$$F_{start}(x) = W'_{start}(GeLU(W_{start}x))$$

$$p_{start}(t_i) = \sigma(F_{start}(x_i))$$

with W'_{start}, W_{start} being the learnable parameters, and σ the sigmoid function. For each start of a mention t_s, i.e., those tokens having $p_{start}(t_s) > 0.5$, we then compute the probability of its subsequent tokens t_j, with $s \leq j$, to be the end of a mention that starts with t_s.

We follow the same process as that of the mention start classification, but we condition the prediction on the starting token by concatenating the start, x_s, and end, x_j, hidden representations before the linear classifier:

$$F_{end}(x, x') = W'_{end}(GeLU(W_{end}[x, x']))$$

$$p_{end}(t_j|t_s) = \sigma(F_{end}(x_s, x_j))$$

with W'_{end}, W_{end} being learnable parameters. This formulation handles overlapping mentions since, for each start t_s, we can find multiple ends t_e (i.e., those that have $p_{end}(t_j|t_s) > 0.5$).

We note that previous works already adopted a linear layer to compute start and end mention scores for each possible mention, i.e., s2e-coref (Kirstain et al., 2021), and LingMess (Otmazgin et al., 2023). However, our mention extraction technique differs from previous approaches since i) we produce two probabilities ($0 < p < 1$) instead of two unbounded scores and ii) we use the computed start probability to filter out possible mentions, which reduces by a factor of 9 the number of mentions considered compared to existing Coarse-to-Fine systems (Table 3.1, first row).

Mention Regularization

To further reduce the computation demand of this process, in the Maverick Pipeline, we use the end-of-sentence (EOS) mention regularization strategy: after extracting the span start, we consider only the tokens up to the nearest EOS as possible

mention end candidates (all the well-established Coreference Resolution datasets are sentence-split). Since annotated mentions never span across sentences, EOS mention regularization prunes the number of mentions considered without any loss of information.

While this heuristic was initially introduced in the implementation of Lee et al. (2018), all the recent Coarse-to-Fine have abandoned it in favor of the maximum span-length regularization, which is a manually-set hyperparameter that regulates a threshold to filter out spans that exceed a certain length. This implies a large overhead of unnecessary computations and introduces a structural bias that does not consider long mentions that exceed a fixed length. In fact, this filters out 196 correctly annotated spans when training on OntoNotes. With Maverick, we not only reintroduce the EOS mention regularization, but we also study its contribution in terms of efficiency, as reported in Table 3.1, second row.

Mention Pruning

After the mention extraction step, as a result of the Maverick Pipeline, we consider an 18x lower number of candidate mentions for the successive mention clustering phase (cf. Table 3.1). This step consists of computing, for each mention, the probability of all its antecedents being in the same cluster, incurring a quadratic computational cost. Within the Coarse-to-Fine formulation, this high computational cost is mitigated by considering only the top k mentions according to their probability score, where k is a manually set hyperparameter. Since after our mention extraction step we obtain probabilities for a very concise number of mentions, we consider only mentions classified as probable candidates (i.e., those with $p_{end} > 0.5$ and $p_{start} > 0.5$), reducing the number of mention pairs considered by a factor of 10. We highlight that this is fundamentally different from setting a manual threshold for top-k since it is equivalent to selecting the most probable class, i.e., the ones that the model classifies as candidate mentions. In Table 3.1, we compare the previous Coarse-to-Fine formulation with the new Maverick Pipeline.

	Coarse-to-Fine	Maverick	Δ
Ment. Extraction	Enumeration	(i) Start-End	
	183,577	20,565	-8,92x
(+) Regularization	(+) Span-length	(ii) (+) EOS	
	14,265	777	-18,3x
Ment. Clustering	Top-k	(iii) Pred-only	
	29,334	2,713	-10,81x

Table 3.1. Comparison between the Coarse-to-Fine pipeline and the Maverick Pipeline in terms of the average number of mentions considered in the mention extraction step (top) and the average number of mention pairs considered in the mention clustering step (bottom). The statistics are computed on the OntoNotes validation set, and refer to the hyperparameters proposed in Lee et al. (2018), which were unchanged by subsequent Coarse-to-Fine works, i.e., span-len = 30, top-k = 0.4.

3.2.2 Mention Clustering

As a result of the Maverick Pipeline, we obtain a set of candidate mentions $M = (m_1, m_2, \ldots, m_l)$, for which we propose three different clustering techniques: $\text{Maverick}_{\text{s2e}}$ and $\text{Maverick}_{\text{mes}}$, which use two well-established Coarse-to-Fine mention-antecedent techniques, and $\text{Maverick}_{\text{incr}}$, which adopts a novel incremental technique that leverages a light transformer architecture.

Mention-Antecedent models

$\text{Maverick}_{\text{s2e}}$ The first proposed model, $\text{Maverick}_{\text{s2e}}$, adopts an equivalent mention clustering strategy to Kirstain et al. (2021): given a mention $m_i = (x_s, x_e)$ and its antecedent $m_j = (x_{s'}, x_{e'})$, with their start and end token hidden states, we use two fully-connected layers to model their corresponding representations:

$$F_s(x) = W'_s(GeLU(W_s x))$$

$$F_e(x) = W'_e(GeLU(W_e x))$$

We then calculate their probability to be in the same cluster as:

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

with $W_{ss}, W_{ee}, W_{se}, W_{es}$ being four learnable matrices and W_s, W'_s, W_e, W'_e the learnable parameters of the two fully-connected layers.

Maverick$_{\text{mes}}$ A similar formulation is adopted in Maverick$_{\text{mes}}$, where, instead of using only one generic mention-pair scorer, we use 6 different scorers that handle linguistically motivated categories, as introduced by Otmazgin et al. (2023). We detect which category k a pair of mentions m_i and m_j belongs to (e.g., if m_i is a pronoun and m_j is a proper noun, the category will be PRONOUN-ENTITY) and use a category-specific scorer to compute p_c.
We use the same set of categories of Otmazgin et al. (2023), namely:

1. PRON-PRON-C. Compatible pronouns based on their attributes such as gender or number (e.g. *(I, I), (I, my) (she, her)*).
2. PRON-PRON-NC, Incompatible pronouns (e.g. *(I, he), (She, my), (his, her)*)
3. ENT-PRON. Pronoun and non-pronoun (e.g. *(George, he), (CNN, it), (Tom Cruise, his)*)
4. MATCH. Non-pronoun spans with the same content words (e.g. *Italy, Italy*)
5. CONTAINS. One contains the other (e.g. *(Barack Obama, Obama)*)
6. OTHER. The Other pairs.

To detect pronouns, we use string match with a full list of English pronouns.

To cluster mentions, we dedicate a mention-pair scorer for each of those categories. Concretely, for the mention $m_i = (x_s^i, x_e^i)$ and its antecedent $m_j = (x_s^j, x_e^j)$, with their start and end token hidden states, we first detect their category

k_g using pattern matching on their spans of texts. Then we compute their start and end representations, using the specific fully connected layers for the category k_g:

$$F_s^{k_g}(x) = W_{k_g,s}^2(GeLU(W_{k_g,s}^1 x))$$

$$F_e^{k_g}(x) = W_{k_g,e}^2(GeLU(W_{k_g,e}^1 x))$$

The probability $P_c^{k_g}$ of m_i and $m_j$is then calculated as:

$$\begin{aligned} P_c^{k_g}(m_i, m_j) = \sigma(&F_s^{k_g}(x_s^i) \cdot \boldsymbol{W_{ss}} \cdot F_s^{k_g}(x_s^j) + \\ &F_e^{k_g}(x_e^i) \cdot \boldsymbol{W_{ee}} \cdot F_e^{k_g}(x_e^j) + \\ &F_s^{k_g}(x_s^i) \cdot \boldsymbol{W_{se}} \cdot F_e^{k_g}(x_e^j) + \\ &F_e^{k_g}(x_e^i) \cdot \boldsymbol{W_{es}} \cdot F_s^{k_g}(x_s^j)) \end{aligned}$$

In this way, each mention-pair scorer learns to model the probability for their specific linguistic categories.

Incremental model

Finally, we introduce a novel incremental approach to tackle the mention clustering step, namely Maverick$_{\text{incr}}$, which follows the standard incremental shift-reduce paradigm introduced in Section 2.2.2. Differently from the previous neural incremental techniques (i.e., ICoref (Xia et al., 2020) and longdoc (Toshniwal et al., 2021)) which use a linear classifier to obtain the clustering probability between each mention and a fixed length vector representation of previously built clusters, Maverick$_{\text{incr}}$ leverages a lightweight transformer model to attend to previous clusters, for which we retain the mentions' hidden representations. Specifically, we compute the hidden representations $(h_1, \ldots, h_l)$ for all the candidate mentions in M using a fully-connected layer on top of the concatenation of their start and end token representations. We first assign the first mention m_1 to the first cluster $c_1 = (m_1)$. Then, for each mention $m_i \in M$ at step i, we obtain the probability of m_i being in a certain cluster c_j by encoding h_i with all the representations of the mentions

contained in the cluster c_j using a transformer architecture. We use the first special token ([CLS]) of a single-layer transformer architecture T to obtain the score $S(m_i, c_j)$ of m_i being in the cluster $c_j = (m_f, \ldots, m_g)$ with $f \leq g < i$ as:

$$S(m_i, c_j) = W_c \cdot (ReLU(T_{CLS}(h_i, h_f, \ldots, h_g)))$$

Finally, we compute the probability of m_i belonging to c_j as:

$$p_c(m_i \in c_j | c_j = (m_f, \ldots, m_g)) = \sigma(S(m_i, c_j))$$

We calculate this probability for each cluster c_j up to step i. We assign the mention m_i to the most probable cluster c_j having $p_c(m_i \in c_j) > 0.5$ if one exists, or we create a new singleton cluster containing m_i.

3.2.3 Training

To train any Maverick model, we optimize the sum of three binary cross-entropy losses:

$$L_{coref} = L_{start} + L_{end} + L_{clust}$$

Our loss formulation differs from previous transformer-based Coarse-to-Fine approaches, which adopt the marginal log-likelihood to optimize the mention to antecedent score (Lee et al., 2018; Kirstain et al., 2021). Since their formulation "makes learning slow and ineffective, especially for mention detection" (Zhang et al., 2018), we directly optimize both mention extraction and mention clustering with a multitask approach. L_{start} and L_{end} are the start loss and end loss, respectively, of the mention extraction step, and are defined as:

$$\begin{aligned} L_{start} = \sum_{i=1}^{N} & -(y_i \log(p_{start}(t_i)) + \\ & (1 - y_i) \log(1 - p_{start}(t_i))) \\ L_{end} = \sum_{s=1}^{S} \sum_{j=1}^{E_s} & -(y_i \log(p_{end}(t_j | t_s)) + \\ & (1 - y_i) \log(1 - p_{end}(t_j | t_s))) \end{aligned}$$

where N is the sequence length, S is the number of starts, E_s is the number of possible ends for a start s and $p_{start}(t_i)$ and $p_{end}(t_j|t_s)$ are those defined in Section 3.2.1.

Finally, L_{clust} is the loss for the mention clustering step. Since we experiment with two different mention clustering formulations, we use a different loss for each clustering technique, namely L^{ant}_{clust} for the mention-antecedent models, i.e., $\text{Maverick}_{\text{s2e}}$ and $\text{Maverick}_{\text{mes}}$, and L^{incr}_{clust} for the incremental model, i.e., $\text{Maverick}_{\text{incr}}$:

$$L^{ant}_{clust} = \sum_{i=1}^{|M|}\sum_{j=1}^{|M|} -(y_i \log(p_c(m_i|m_j)) + (1-y_i)\log(1-p_c(m_i|m_j)))$$

$$L^{incr}_{clust} = \sum_{i=1}^{|M|}(\sum_{j=1}^{C_i} -(y_i \log(p_c(m_i \in c_j)) + (1-y_i)\log(1-p_c(m_i \in c_j))))$$

where $|M|$ is the number of extracted mentions, C_i is the set of clusters created up to step i, and $p_c(m_i|m_j)$ and $p_c(m_i \in c_j)$ are defined in Section 3.2.2.

All the models we introduce are trained using teacher forcing. In particular, in the mention token end classification step, we use gold start indices to condition the end tokens prediction, and, for the mention clustering step, we consider only gold mention indices. For $\text{Maverick}_{\text{incr}}$, at each iteration, we compare each mention only to previous gold clusters.

3.3 Experiments Setup

We now present the datasets and settings used for our experiments, which were briefly introduced in Section 2.2.1. After that, we will include comparison systems and the Maverick models' setup.

3.3.1 Datasets

We train and evaluate all the comparison systems on three Coreference Resolution datasets:

Dataset	# Train	# Dev	# Test	Tokens	Mentions	% Sing
OntoNotes	2802	343	348	467	56	0
LitBank	80	10	10	2105	291	19.8
PreCo	36120	500	500	337	105	52.0
GAP	-	-	2000	95	3	-
WikiCoref	-	-	30	1996	230	0

Table 3.2. Dataset statistics: number of documents in each dataset split, average number of words and mentions per document, and singletons percentage.

OntoNotes (Pradhan et al., 2012), proposed in the CoNLL-2012 shared task, is the de facto standard dataset used to benchmark Coreference Resolution systems. It consists of documents that span seven distinct genres, including full-length documents (broadcast news, newswire, magazines, weblogs, and Testaments) and multiple speaker transcripts (broadcast and telephone conversations).

LitBank (Bamman et al., 2020) contains 100 literary documents typically used to evaluate long-document Coreference Resolution.

PreCo (Chen et al., 2018) is a large-scale dataset that includes reading comprehension tests for middle school and high school students.

Notably, as we already detailed in Section 2.2.1, both LitBank and PreCo have different annotation guidelines compared to OntoNotes, and provide annotation for singletons (i.e., single-mention clusters).
Furthermore, we evaluate models trained on OntoNotes on three out-of-domain datasets:

- **GAP** (Webster et al., 2018) contains sentences in which, given a pronoun, the model has to choose between two candidate mentions.
- **LitBank$_{\mathbf{ns}}$** and **PreCo$_{\mathbf{ns}}$**, the datasets' test-set where we filter out singleton annotations.
- **WikiCoref** (Ghaddar and Langlais, 2016), which contains Wikipedia texts, including documents with up to 9,869 tokens.

The statistics of the adopted datasets are shown in Table 3.2.

3.3.2 Comparison Systems

Discriminative Among the discriminative systems, we consider c2f-coref (Joshi et al., 2020) and s2e-coref (Kirstain et al., 2021), which build upon the Coarse-to-Fine formulation and adopt different document encoders. We also report the results of LingMess (Otmazgin et al., 2023), which is the previous best encoder-only solution, and f-coref (Otmazgin et al., 2022), which is a distilled version of LingMess. Furthermore, we include CorefQA (Wu et al., 2020), which casts Coreference as extractive Question Answering, and wl-coref (Dobrovolskii, 2021), which first predicts coreference links between words, then extracts mention spans. Finally, we report the results of incremental systems, such as ICoref (Xia et al., 2020) and longdoc (Toshniwal et al., 2021).

Sequence-to-Sequence We compare our models with TANL (Paolini et al., 2021) and ASP (Liu et al., 2022), which frame Coreference Resolution as an autoregressive structured prediction. We also include Link-Append (Bohnet et al., 2023), a transition-based system that builds clusters with a multi-pass Sequence-to-Sequence architecture. Finally, we report the results of seq2seq (Zhang et al., 2023), a model that learns to generate a sequence with Coreference Resolution labels.

3.3.3 Maverick Setup

Each of the Maverick models proposed in this chapter uses DeBERTa-v3 (He et al., 2023) as the document encoder. We use DeBERTa because it can model very long input texts effectively (He et al., 2021), since its attention mechanism enables its input length to grow linearly with the number of its layers. Moreover, compared to the LongFormer, which was previously adopted by several token-level systems, DeBERTa ensures a larger input max sequence length (e.g., $\text{DeBERTa}_{\text{large}}$ can handle sequences up to 24,528 tokens while LongFormer only 4096) and has shown better performances empirically in our experiments on the OntoNotes dataset. On the other hand, using DeBERTa to encode long documents is computationally expensive because its attention mechanism incurs a quadratic computational complexity. Whereas this further increases the computational cost of traditional Coarse-to-Fine systems, the

Maverick Pipeline enables us to train models that leverage DeBERTa$_{\text{large}}$ on the OntoNotes dataset, without any performance-lowering pruning heuristic. To train our models, we use Adafactor (Shazeer and Stern, 2018) as our optimizer, with a learning rate of 3e-4 for the linear layers, and 2e-5 for the pre-trained encoder. We perform all our experiments within an academic budget, i.e., a single RTX 4090, which has 24GB of VRAM.

When training on the above-mentioned datasets, we have the following setups:

- **OntoNotes** contains several items of metadata information for each document, such as genre, speakers, and constituent graphs. Following previous works, we incorporate the speaker's name into the text whenever there is a change in speakers for datasets that include this metadata.
- **LitBank** contains 100 literary documents and is available in 10 different cross-validation folds. Our train, dev, and test splits refer to the first cross-validation fold, LB$_0$. We report comparison systems results on the same splits. Since training DeBERTa$_{\text{large}}$ is particularly computationally expensive, we train on LitBank by splitting in half each LitBank training document.
- The authors of **PreCo** 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.

3.4 Results

3.4.1 In-domain evaluation

English OntoNotes

We report in Table 3.3 the average CoNLL-F1 score of the comparison systems trained on the English OntoNotes, along with their underlying pre-trained language models and total parameters. Compared to previous discriminative systems, we report gains of +2.2 CoNLL-F1 points over LingMess, the best encoder-only model. Interestingly, we even outperform CorefQA, which uses additional Question Answering training data.

Model	LM	Avg. F1	Params	Training		Inference	
				Time	Hardware	Time	Mem.
Discriminative							
c2f-coref (Joshi et al., 2020)	$\text{SpanBERT}_{\text{large}}$	79.6	370M	-	1x32G	50s	11.9
ICoref (Xia et al., 2020)	$\text{SpanBERT}_{\text{large}}$	79.4	377M	40h	1x1080TI-12G	38s	2.9
CorefQA (Wu et al., 2020)	$\text{SpanBERT}_{\text{large}}$	83.1*	740M	-	1xTPUv3-128G	-	-
s2e-coref (Kirstain et al., 2021)	$\text{LongFormer}_{\text{large}}$	80.3	494M	-	1x32G	17s	3.9
longdoc (Toshniwal et al., 2021)	$\text{LongFormer}_{\text{large}}$	79.6	471M	16h	1xA6000-48G	25s	2.1
wl-coref (Dobrovolskii, 2021)	$\text{RoBERTa}_{\text{large}}$	81.0	360M	5h	1xRTX8000-48G	11s	2.3
f-coref (Otmazgin et al., 2022)	DistilRoBERTa	78.5*	91M	-	1xV100-32G	3s	1.0
LingMess (Otmazgin et al., 2023)	$\text{LongFormer}_{\text{large}}$	81.4	590M	23h	1xV100-32G	20s	4.8
Sequence-to-Sequence							
ASP (Liu et al., 2022)	T5-base	76.6	220M	-	1xA100-40G	-	-
	T5-large	79.3	770M	-	1xA100-40G	-	-
	FLAN-T5L	80.2	770M	-	1xA100-40G	-	-
	T0-3B	82.3	3B	-	2xA100-40G	-	-
	FLAN-T5xl	82.2	3B	-	2xA100-40G	-	-
	FLAN-T5xxl	82.5	11B	45h	6xA100-80G	20m	-
Link-Append (Bohnet et al., 2023)	mT5xl	78.0^d	3B	-	128xTPUv4-32G	-	-
	T5xxl	82.7^d	11B	-	128xTPUv4-32G	-	-
	mT5xxl	83.3	13B	48h	128xTPUv4-32G	30m	-
seq2seq (Zhang et al., 2023)	T5-base	76.2^d	220M	-	8xA100-40G	-	-
	T5-large	77.2^d	770M	-	8xA100-40G	-	-
	T5-3B	81.6^d	3B	-	8xA100-40G	-	-
	FLAN-T5xl	82.5^d	3B	-	8xA100-40G	-	-
	T0-3B	82.5	3B	-	8xA100-40G	-	-
	FLAN-T5xxl	82.9^d	11B	-	8xA100-80G	-	-
	T0-11B	83.2	11B	-	8xA100-80G	40m	-
Ours (Discriminative)							
$\text{Maverick}_{\text{s2e}}$	$\text{DeBERTa}_{\text{base}}$	81.1	192M	7h	1xRTX4090-24G	6s	1.8
	$\text{DeBERTa}_{\text{large}}$	83.4	449M	14h	1xRTX4090-24G	13s	4.0
$\text{Maverick}_{\text{incr}}$	$\text{DeBERTa}_{\text{base}}$	81.0	197M	21h	1xRTX4090-24G	22s	1.8
	$\text{DeBERTa}_{\text{large}}$	83.5	452M	29h	1xRTX4090-24G	29s	3.4
$\text{Maverick}_{\text{mes}}$	$\text{DeBERTa}_{\text{base}}$	81.4	223M	7h	1xRTX4090-24G	6s	1.9
	$\text{DeBERTa}_{\text{large}}$	**83.6**	504M	14h	1xRTX4090-24G	14s	4.0

Table 3.3. Results on the OntoNotes benchmark. We report the Avg. CoNLL-F1 score, the number of parameters, the training time, and the hardware used to train each model. Inference time (sec) and memory (GiB) were calculated on an RTX4090. For Sequence-to-Sequence models we include statistics that are reported in the original papers, since we could not run models locally. (*) indicates models trained on additional resources. (^{d}) indicates scores obtained on the dev set; however, Maverick systems always perform better on the dev set than on the test sets. Missing values (-) are not reported in the original paper, and it is not feasible to reproduce them using our limited hardware resources.

Concerning Sequence-to-Sequence approaches, we report extensive improvements over systems with a similar amount of parameters compared to our large models (500M): we obtain +3.4 points compared to ASP (770M), and the gap is even wider when taking into consideration Link-Append (3B) and seq2seq (770M), with +6.4 and +5.6, respectively. Most importantly, Maverick models surpass the performance of all Sequence-to-Sequence transformers even when they have several billion parameters. Among our proposed methods, $\text{Maverick}_{\text{mes}}$ shows the best performance, setting a new state of the art with a score of 83.6 CoNLL-F1 points on the OntoNotes benchmark.

Model	PreCo	LitBank
longdoc (Toshniwal et al., 2021)	87.8	77.2
seq2seq (Zhang et al., 2023)	**88.5**	77.3
$\text{Maverick}_{\text{s2e}}$	87.2	77.6
$\text{Maverick}_{\text{incr}}$	88.0	**78.3**
$\text{Maverick}_{\text{mes}}$	87.4	78.0

Table 3.4. Results of the compared systems on the PreCo and LitBank test-sets in terms of CoNLL-F1 score.

PreCo and LitBank

We further validate the robustness of the Maverick framework by training and evaluating systems on the PreCo and LitBank datasets. As reported in Table 3.4, our models show superior performance when dealing with long documents in a data-scarce setting such as the one LitBank poses. On this dataset, $\text{Maverick}_{\text{incr}}$ achieves a new state-of-the-art score of 78.3, and gains +1.0 CoNLL-F1 points compared with seq2seq. On PreCo, $\text{Maverick}_{\text{incr}}$ outperforms longdoc, but seq2seq still shows slightly better performance. This is mainly due to the high presence of singletons in PreCo (52% of all the clusters). Our systems, using a mention extraction technique that favors precision rather than recall, are penalized compared to high recall systems such as seq2seq. Among our systems, $\text{Maverick}_{\text{incr}}$, leveraging its hybrid architecture, performs better on both PreCo and LitBank.

3.4.2 Out-of-Domain Evaluation

In Table 3.5, we report the performance of Maverick systems along with LingMess, the best encoder-only model, when dealing with out-of-domain texts, that is, when they are trained on OntoNotes and tested on other datasets. First of all, we report considerable improvements on the GAP test set, obtaining a +1.2 F1 score compared to the previous state of the art. We also test models on WikiCoref, PreCo_{ns} and $\text{LitBank}_{\text{ns}}$ (Section 3.3.1). However, since the span annotation guidelines of these corpora differ from the ones used in OntoNotes, in Table 3.5 we also report the performance using gold mentions, i.e., skipping the mention extraction step (gold column).[1] On the WikiCoref benchmark, we achieve a new state-of-the-art score of

[1]We do not include autoregressive models because none of the original articles report scores on out-of-domain datasets. We also could not test those models because they do not provide the code to

Model	GAP	WikiCoref		PreCo_{ns}		$\text{LitBank}_{\text{ns}}$	
		sys.	gold	sys.	gold	sys.	gold
LingMess	89.6	63.0	76.6	65.1	80.6	64.4	73.9
$\text{Maverick}_{\text{s2e}}$	91.1	**67.2**	81.5	**67.2**	**87.9**	64.8	83.1
$\text{Maverick}_{\text{incr}}$	**91.2**	66.8	**82.4**	66.1	86.5	**65.4**	**84.0**
$\text{Maverick}_{\text{mes}}$	91.1	66.8	82.1	66.1	86.9	65.1	82.8

Table 3.5. Comparison between LingMess and Maverick systems on GAP, WikiCoref, PreCo_{ns} $\text{LitBank}_{\text{ns}}$. We report scores using systems prediction (sys..) or passing gold mentions (gold).

67.2 CoNLL-F1, with an improvement of +4.2 points over the previous best score obtained by LingMess. On the same dataset, when using pre-identified mentions, the gap increases to +5.8 CoNLL-F1 points (76.6 vs 82.4). In the same setting, our models obtain up to +7.3 and +10.1 CoNLL-F1 points on Preco_{ns} and $\text{LitBank}_{\text{ns}}$, respectively, compared to LingMess. These results suggest that the Maverick training strategy makes this model more suitable when dealing with pre-identified mentions and out-of-domain texts. This further increases the potential benefits that Maverick systems can bring to many downstream applications that exploit coreference as an intermediate layer, such as Entity Linking (Rosales-Méndez et al., 2020) and Relation Extraction (Xiong et al., 2023; Zeng et al., 2023), where the mentions are already identified. Among our models, on $\text{LitBank}_{\text{ns}}$ and WikiCoref, $\text{Maverick}_{\text{incr}}$ outperforms $\text{Maverick}_{\text{mes}}$ and $\text{Maverick}_{\text{s2e}}$, confirming the superior capabilities of the incremental formulation in the long-document setting. Finally, we highlight that the performance gap between using gold mentions and performing full Coreference Resolution is wider when tested on out-of-domain datasets (on average +17%) compared to testing it directly on OntoNotes (83.6 vs 93.6, +10%, cf. Section 3.4.5). This result, obtained on three different out-of-domain datasets, suggests that the difference in annotation guidelines considerably contributes to lowering the OOD performances (-7%).

perform mention clustering alone, and performing it with such approaches is not as straightforward as in encoder-only models.

3.4.3 Speed and Memory Usage

In Table 3.3, we include details regarding the training time and the hardware used by each comparison system, along with the measurement of the inference time and peak memory usage on OntoNotes' validation set. Compared to Coarse-to-Fine models, which require 32GB of VRAM, we can train Maverick systems under 18GB. At inference time both Maverick$_{\text{mes}}$ and Maverick$_{\text{s2e}}$, exploiting DeBERTa$_{\text{large}}$, achieve competitive speed and memory consumption compared to wl-coref and s2e-coref. Furthermore, when adopting DeBERTa$_{base}$, Maverick$_{\text{mes}}$ proves to be the most efficient approach[2] among those directly trained on OntoNotes, while, at the same time, attaining performances that are equal to the previous best encoder-only system, LingMess. The only system that shows better inference speed is f-coref, but at the cost of lower performance (-3.0 points).

Compared to the previous Sequence-to-Sequence state-of-the-art approach, Link-Append, we train our models with 175x less memory requirements. Comparing inference time is more complicated, since we could not run models on our memory-constrained budget. For this reason, we report the inference times from the original articles, and hence times achieved with their high-resource settings. Interestingly, we report as much as 170x faster inference compared to seq2seq, which exploits parallel inference on multiple GPUs, and 85x faster when compared to the more efficient ASP. Among Maverick models, Maverick$_{\text{incr}}$ is notably slower both in inference and training time, as it incrementally builds clusters using multiple steps.

3.4.4 Ablation study

In Table 3.6, we compare Maverick$_{\text{s2e}}$ and Maverick$_{\text{mes}}$ models with s2e-coref and LingMess, respectively, using different pre-trained encoders. Interestingly, when using DeBERTa, Maverick systems not only achieve better speed and memory efficiency but also obtain higher performance compared to the previous systems. When using the LongFormer, instead, their scores are in the same ballpark, showing empirically that the Maverick training procedure better exploits the capabilities of DeBERTa. To test the benefits of our novel incremental formulation, Maverick$_{\text{incr}}$,

[2] In terms of inference peak memory usage and speed.

Model	LM	Score
Maverick$_{\text{s2e}}$		
Maverick$_{\text{s2e}}$	DeBERTa$_{\text{base}}$	81.0
s2e-coref$_{\text{t}}$	DeBERTa$_{\text{base}}$	78.3
Maverick$_{\text{s2e}}$	LongFormer$_{\text{large}}$	80.6
s2e-coref	LongFormer$_{\text{large}}$	80.3
Maverick$_{\text{mes}}$		
Maverick$_{\text{mes}}$	DeBERTa$_{\text{base}}$	81.4
LingMess$_{\text{t}}$	DeBERTa$_{\text{base}}$	78.6
Maverick$_{\text{mes}}$	LongFormer$_{\text{large}}$	81.0
LingMess	LongFormer$_{\text{large}}$	81.4
Maverick$_{\text{incr}}$		
Maverick$_{\text{incr}}$	DeBERTa$_{\text{large}}$	83.5
Maverick$_{\text{prev-incr}}$	DeBERTa$_{\text{large}}$	79.6

Table 3.6. Comparison between Maverick models and previous techniques. LingMess$_{\text{t}}$ and s2e-coref$_{\text{t}}$ are trained using their official scripts. We use DeBERTa$_{\text{base}}$ because the DeBERTa$_{\text{large}}$ could not fit our hardware when training comparison systems.

we also implement a Maverick model with the previously adopted incremental method used in longdoc and ICoref (incremental models, Section 2.2.2), which we call Maverick$_{\text{prev-incr}}$. Compared to the previous formulation, we report an increase in score of +3.9 CoNLL-F1 points. The improvement demonstrates that exploiting a transformer architecture to attend to all the previously clustered mentions is beneficial and enables the future usage of hybrid architectures when needed.

As a further analysis of whether the efficiency improvements of our systems stem from using DeBERTa or are attributable to the Maverick Pipeline, we compared the speed and memory occupation of a Maverick system using as an underlying encoder either DeBERTa$_{\text{large}}$ or LongFormer$_{\text{large}}$. Our experiments show that using DeBERTa leads to an increase of +77% of memory space and +23% of time to complete an epoch when training on OntoNotes. An equivalent measurement, attributable to the quadratic memory attention mechanism of DeBERTa, was observed for the inference time and memory occupation on the OntoNotes test set.

3.4.5 Error Analysis

To better understand the quality of Maverick predictions, we conduct an error analysis on our best system trained on OntoNotes, Maverick$_{\text{mes}}$. In Table 3.7, we

System	Ment. Clustering	Ment. Extraction
$\text{Maverick}_{\text{s2e}}$	89.4	93.5
$\text{Maverick}_{\text{incr}}$	89.2	94.2
$\text{Maverick}_{\text{mes}}$	89.6	93.7

Table 3.7. Mention extraction (F1) and mention clustering (CoNLL-F1) scores on the OntoNotes validation set.

report the score of performing only mention extraction (F1) or mention clustering with gold mention (CoNLL-F1) with our systems. Our results highlight that our models have strong capabilities of clustering pre-identified mentions, but limited performance in the identification of correct spans. We investigated this phenomenon by conducting a qualitative evaluation of the outputs of our best system, $\text{Maverick}_{\text{mes}}$, and found out that OntoNotes contains several annotation errors.

We report examples of errors in Table 3.8. The main inconsistency we found in the gold test set is that many documents have incomplete annotations, which directly correlates with the mention extraction error.

3.5 Summary

In this chapter, we challenge the recent trends of adopting large autoregressive generative models to solve the Coreference Resolution task. To do so, we proposed Maverick, a new framework that enables fast and memory-efficient Coreference Resolution while obtaining state-of-the-art results. This demonstrates that the large computational overhead required by Sequence-to-Sequence approaches is unnecessary. Indeed, in our experiments, Maverick systems demonstrated that they can outperform large generative models and improve the speed and memory usage of previous best-performing encoder-only approaches. Furthermore, we introduced Maverick_{incr}, a robust multi-step incremental technique that obtains higher performance in out-of-domain settings. By releasing our systems, we make state-of-the-art models usable by a larger portion of users in different scenarios and potentially improve downstream applications. In this thesis, Maverick has a central role, as in the next chapters, we will use it to tag new resources and as an underlying technique for other, more robust models for long-document processing.

Type	Text
Ex. 1	
Gold	Nine people were injured in Gaza when gunmen [**opened**]$_1$[**fire**]$_2$on an Israeli bus. The passengers were off - duty Israeli security workers. Witnesses say [**the shots**]$_2$came from [**the Palestinian international airport**]$_3$. Israeli Prime Minister Ehud Barak [**closed**]$_4$down [**the two - year - old airport**]$_3$in response to [**the incident**]$_1$. [**Palestinians**]$_5$criticized [**the move**]$_4$. [**hey**]$_5$regard [**the airport**]$_3$as a symbol of emerging statehood.
Output	[**Nine people**]$_1$were injured in Gaza when gunmen opened fire on an Israeli bus. [**The passengers**]$_1$were off - duty Israeli security workers. Witnesses say the shots came from [**the Palestinian international airport**]$_2$. Israeli Prime Minister Ehud Barak [**closed**]$_3$down [**the two - year - old airport**]$_2$in response to the incident. [**Palestinians**]$_4$criticized [**the move**]$_3$. [**They**]$_4$regard [**the airport**]$_2$as a symbol of emerging statehood.
Ex. 2	
Gold	[**Mr. Seelenfreund**]$_1$is [**executive vice president and chief financial officer of** [**McKesson**]$_3$]$_2$- and will continue in [**those roles**]$_2$. [**PCS**]$_4$also named Rex R. Malson, 57, executive vice president at McKesson,- as a director, filling the seat vacated by Mr. Field. Messrs. Malson and Seelenfreund are directors of [**McKesson, which has an 86% stake in** [**PCS**]$_4$]$_3$.
Output	[**Mr. Seelenfreund**]$_1$is [**executive vice president and chief financial officer of** [**McKesson**]$_3$]$_2$ and will continue in [those roles]$_2$. [**PCS**]$_4$also named [**Rex R. Malson, 57, executive vice president at** [**McKesson**]$_3$,]$_5$- as a director, filling the seat vacated by Mr. Field. Messrs. [**Malson**]$_5$and [**Seelenfreund**]$_1$are directors of [**McKesson, which has an 86 % stake in** [**PCS**]$_4$]$_3$.
Ex. 3	
Gold	The Second U.S. Circuit Court of Appeals opinion in the Arcadian Phosphate case - did not repudiate the position [**Pennzoil Co.**]$_1$took in [**its**]$_1$dispute with [**Texaco**]$_2$, - contrary to your Sept. 8 article " Court Backs [**Texaco**]$_2$'s View in [**Pennzoil**]$_1$Case – Too Late. " The fundamental rule of contract law applied to [**both cases**]$_3$was that courts will not enforce - [**agreements to** [**which**]$_4$**the parties did not intend to be bound**]$_4$. In the Pennzoil / Texaco litigation, [**the courts**]$_5$found [**Pennzoil**]$_1$and Getty Oil intended to be bound; in Arcadian Phosphates [**they**]$_5$found there was no intention to be bound.
Output	The Second U.S. Circuit Court of Appeals opinion in [**the Arcadian Phosphate case**]$_1$ - did not repudiate the position [**Pennzoil Co.**]$_2$took in [[[**its**]$_2$**dispute with** [**Texaco**]$_4$]$_3$**, -** **contrary to your Sept. 8 article " Court Backs** [**Texaco 's**]$_4$**View in** [[**Pennzoil**]$_2$**Case**]$_3$]$_3$– Too Late . " [[**The fundamental rule of contract law**]$_5$**applied to both cases**]$_5$was that courts will not enforce - agreements to which the parties did not intend to be bound. In [**the** [**Pennzoil**]$_2$ **/** [**Texaco**]$_4$**litigation**]$_3$, [**the courts**]$_6$found [**Pennzoil**]$_2$and Getty Oil intended to be bound; in [**Arcadian Phosphates**]$_1$[**they**]$_6$found there was no intention to be bound.
Ex. 4	
Gold	... [**Harry**]$_1$has avoided all that by living in a Long Island suburb with [**his**]$_1$wife, who 's so addicted to soap operas and mystery novels she barely seems to notice when [**her husband**[disappears for drug - seeking forays into Manhattan.
Output	... [**Harry**]$_1$has avoided all that by living in a Long Island suburb with [[**his**]$_1$**wife,** **who 's so addicted to soap operas and mystery novels** [**she**]$_2$**barely seems to notice when** [[**her**]$_2$**husband**]$_1$**disappears for drug - seeking forays into Manhattan**]$_2$.

Table 3.8. OntoNotes test set annotation errors examples.

Chapter 4

Extending Coreference Resolution on Full Narrative Books

Abstract

In the previous chapter, we detailed Maverick, an efficient and accurate method to solve the Coreference Resolution task, and we highlighted the current need for efficient techniques. However, Maverick and other current systems are typically evaluated on benchmarks containing small- to medium-scale documents, and when it comes to evaluating long texts, existing benchmarks remain limited in length and do not adequately assess system capabilities at the book scale, i.e., when coreferring mentions span hundreds of thousands of tokens.

To fill this gap, in this chapter, we first introduce a novel automatic pipeline that produces high-quality CR annotations on full narrative texts. Then, we adopt this pipeline to create the first book-scale coreference benchmark, BookCoref, with an average document length of more than 200,000 tokens. Moreover, we perform a series of experiments to show the robustness of our automatic procedure and demonstrate the value of our resource, which enables current long-document coreference systems to gain up to +20 CoNLL-F1 points when evaluated on full books. Finally, we report on the new challenges introduced by this unprecedented book-scale setting, highlighting that current models fail to deliver the same performance they achieve on smaller documents.

Source: This chapter is based on the ACL 2025 paper: BookCoref: *Coreference Resolution at Book Scale* (Martinelli et al., 2025)
Code: `https://github.com/sapienzanlp/bookcoref`

4.1 Overview

Co-referring mentions are usually found across multiple sentences, and can be located far apart within the same document. For this reason, as the document length increases, so does the difficulty of manually annotating a corpus with coreference relations (Roesiger et al., 2018). This is because humans typically resolve coreference relations incrementally (Altmann and Steedman, 1988b), and therefore – when annotating a coreference mention – they must rely on previously annotated entities and take the preceding context into account (Roesiger et al., 2018). This labor-intensive process results in a higher cost of human annotations of long-form texts. As a result, current benchmarks ease the annotation process by drawing samples from short- or medium-sized textual genres (i.e., news, broadcasts, and magazines), or by artificially shortening documents. For instance, the two most widely used English coreference benchmarks, OntoNotes (Pradhan et al., 2013) and LitBank (Bamman et al., 2020), either split documents into smaller chunks or truncate them to the first 2,000 tokens. For this reason, most CR systems are optimized for shorter texts and struggle to handle longer inputs effectively.

Several recent works, such as LongtoNotes (Shridhar et al., 2023) and MovieCoref (Baruah et al., 2021; Baruah and Narayanan, 2023), attempt to address this gap by introducing manually-annotated long-document CR resources. However, LongtoNotes documents remain limited in length (less than 700 tokens per document), while MovieCoref contains only a small number of full-length screenplays, a niche narrative genre with an atypical textual structure. We argue that the current lack of book-scale benchmarks leaves the many challenges this setting poses, such as efficient processing (Toshniwal et al., 2020, 2021), consistency (Guo et al., 2023), and evaluation (Duron-Tejedor et al., 2023) of Coreference Resolution in long documents, largely unexplored.

To address this limitation, we introduce a highly reliable automatic pipeline to annotate long documents, which we then apply to full-length literary works to create the first book-scale coreference benchmark, BookCoref. Inspired by recent studies that suggest focusing on a smaller yet relevant set of entities (Baruah et al., 2021; Guo et al., 2023; Manikantan et al., 2024), our annotation involves only the

characters that appear in a book, as they are the main agents of fictional stories (Bamman et al., 2013; Roesiger et al., 2018; Labatut and Bost, 2019). As shown in Figure 4.1, BookCoref presents unprecedented long-document characteristics, enabling the study of the coreference phenomena in full narrative books, which was previously impossible.

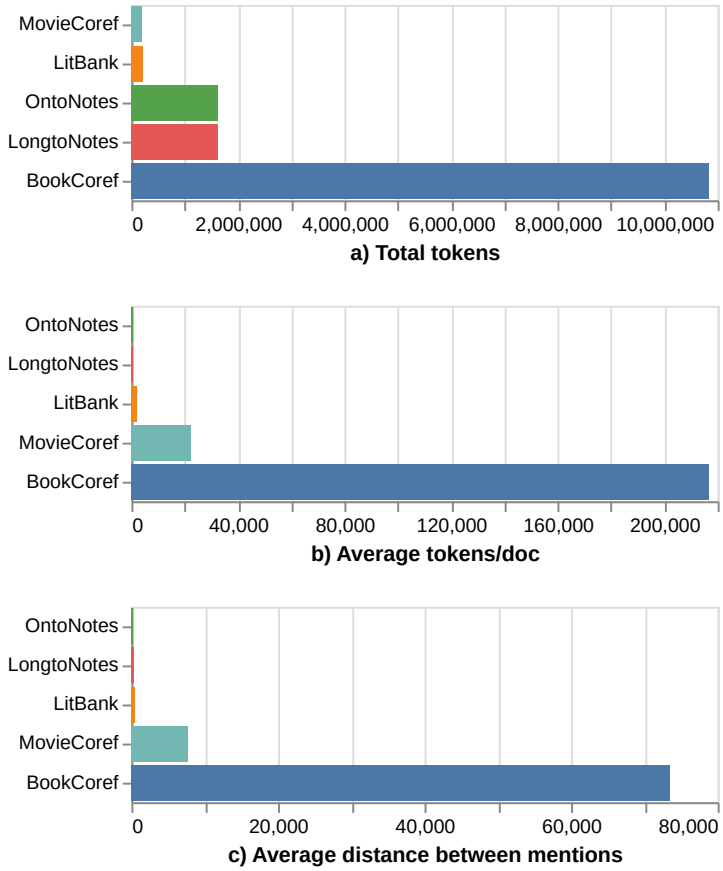


Figure 4.1. Comparison between BookCoref and current long-document resources, measuring: a) the total number of tokens, b) the average number of tokens per document, and c) the micro-average pairwise distance between all co-referring mentions.

To summarize, in this chapter:

- We put forward the BookCoref Pipeline, a novel procedure that enables Coreference Resolution annotation of full documents, and extensively validate its effectiveness.
- We introduce BookCoref, a new book-scale Coreference Resolution dataset with a manually-annotated split, BookCoref$_{GOLD}$, to train and evaluate

systems on fully annotated books.[1]

- We benchmark state-of-the-art Coreference Resolution systems, reporting on the open challenges of this new book-scale setting.

4.2 The BookCoref Corpus

Our automatic pipeline annotates full-text documents starting from their list of characters. Therefore, to produce BookCoref we leverage three main resources: i) Project Gutenberg[2], a collection of openly available literary works; ii) Wikidata, a multilingual knowledge graph containing thousands of books[3]; iii) LiSCU (Brahman et al., 2021), a dataset that collects information from online study guides on the characters appearing in English literary works.

We start by collecting a list of characters, authors, and titles of books available in Wikidata and LiSCU. As these resources do not provide full texts, we search Project Gutenberg for complete books that match the corresponding authors and titles. To ensure high quality, we manually remove erroneous matches and automatically exclude books with fewer than five characters.

Formally, after this step, we obtain a set of full-text books $\mathcal{B}$ where each book $b \in \mathcal{B}$ is paired with a list of character names $\mathcal{C}(b) = \{c_1, c_2, \ldots, c_n\}$. Our corpus contains $|\mathcal{B}| = 53$ books with an average of 27 characters per book, mostly comprising full-length classical novels such as Walter Scott's *Ivanhoe* or Jane Austen's *Pride and Prejudice*. We include in Table 4.1 the final list of books in BookCoref.

4.2.1 Resource Details

The authors of LiSCU (Brahman et al., 2021) do not directly release their data but instead redirect to a GitHub repository to scrape the Internet Archive and obtain the full dataset.[4] We fix broken links, remove some restrictive filters from their codebase that skip minor characters, and run their scripts to save a local copy of LiSCU in a

[1] We release BookCoref on HuggingFace: `https://huggingface.co/datasets/sapienzanlp/bookcoref`.

[2] `https://www.gutenberg.org/`

[3] As of 2024-12-09, Wikidata contains 75,675 book items.

[4] `https://github.com/huangmeng123/lit_char_data_wayback`

	Book name	Gutenberg key	Number of tokens	Number of characters
	The Pilgrim's Progress	7088	30610	45
	The Boxcar Children	42796	30819	11
	Dr. Jekyll and Mr. Hyde	42	31127	8
G	Animal Farm*	-	36504	20
G	Siddhartha	2500	47785	9
	The Awakening	160	60031	26
	Pudd'nhead Wilson	102	63061	20
	O Pioneers!	24	67463	12
	The Hound of the Baskervilles	2852	70582	19
	The Prince and the Pauper	1837	81947	21
	Silas Marner	550	87218	33
	Northanger Abbey	121	92861	17
	My Ántonia	19810	95698	38
	Kidnapped	421	98379	21
	Persuasion	105	99102	20
	Fathers and Sons	47935	99902	27
	The Tale of Genji	66057	106868	20
	Pan Tadeusz	28240	136711	10
	Sense and Sensibility	161	142607	18
	The Mayor of Casterbridge	143	143129	13
G	Pride and Prejudice	1342	146495	38
	The Talisman	1377	155678	10
	Moll Flanders	370	159287	14
	In Search of the Castaways	46597	166845	10
	A Tale of Two Cities	98	170076	14
	The Last of the Mohicans	27681	172400	12
	The Return of the Native	122	174304	18
	The Jungle	140	177580	46
	The Way of All Flesh	2084	185562	39
	Emma	158	190921	24
	The Canterbury Tales	2383	205165	45
	Main Street	543	209394	84
	The Red and the Black	44747	222200	24
	The Man in the Iron Mask	2759	222328	33
	Ivanhoe	82	224176	70
	North and South	4276	224604	11
	Little Women	514	229337	61
	Jane Eyre	1260	231978	34
	Uncle Tom's Cabin	203	233220	25
	The Deerslayer	3285	253815	15
	Pamela	6124	267364	35
	The Three Musketeers	1257	292259	31
	The Woman in White	583	294178	13
	Twenty Years After	1259	315692	20
	Middlemarch	145	378961	35
	The Pickwick Papers	580	388962	88
	Our Mutual Friend	883	417160	17
	Little Dorrit	963	418851	15
	The Brothers Karamazov	28054	435055	12
	Bleak House	1023	437143	66
	Don Quixote	996	473968	31
	War and Peace	2600	675324	32
	Les Miserables	135	689400	45

Table 4.1. List of all books that compose BookCoref, together with their Gutenberg ID, number of tokens, and number of characters. With the color green and the letter G, we identify books that were manually annotated to create BookCoref$_{GOLD}$. *: *Animal Farm* is included from Guo et al. (2024) with no further changes.

local DB. We then export a list of 24,985 tuples, each specifying a character name, the author and title of the book it is linked to, and the internet study guide that reports this information. The total number of books extracted from LiSCU is 1,707, each linked to 14.6 characters on average.

For Wikidata, we derive our own SPARQL query to use three main Wikidata relations: *characters* (P674), which links an item such as a book or a film to characters that appear in it; *author* (P50), which links to the author of a specific work; and *title* (P1476), the published name of a literary work or newspaper article or others. Running this query results in 17,324 distinct (author, title) tuples, each linked on average to 1.54 characters. This highlights some issues of the Wikidata Knowledge Graph, as the character relation has poor coverage, and there is no reliable way to restrict an item identified by an (author, title) tuple to be a work of literary text.

We then attempt to match each (author, title) obtained from the concatenation of LiSCU and Wikidata to a valid book in Project Gutenberg. We use the `gutenbergpy` library[5], which includes functions to search through the whole Project Gutenberg book metadata for an exact token match between our (author, title) tuples and the author and title information in Gutenberg. There is a possibility of multiple (author, title) matches in our character tuples, as there might be some overlap between the sources they are derived from (Wikidata and the three different study guides contained in LiSCU). We simply select the source that contains the largest number of characters for a given book. We manually validate the links between the list of characters of a book and its raw text extracted from Gutenberg, as well as manually cleaning each Gutenberg book to remove headings, placeholders, notes, etc. This leaves us with our base resource: a set of 52 clean books obtained from Project Gutenberg, each linked to a list of characters that appear in the book.

We also include in our corpus *Animal Farm* by George Orwell, as contributed by Guo et al. (2024), resulting in a total number of 53 books.

[5] `https://github.com/raduangelescu/gutenbergpy`

4.3 Automatic Pipeline for Long-document Annotation

We now introduce the BookCoref Pipeline, our proposed automatic annotation procedure. Formally, the BookCoref Pipeline takes as input a book b and its set of character names $\mathcal{C}(b)$ and outputs coreference cluster annotations over the full book for each character name. Notably, although we apply our pipeline to narrative books, it is genre-agnostic and can be exploited for any long document, provided a set of named entities is available. Figure 4.2 summarizes the steps of our BookCoref Pipeline, namely, Cluster Initialization (Section 4.3.1), Cluster Refinement (Section 4.3.2), and Cluster Expansion (Section 4.3.3).

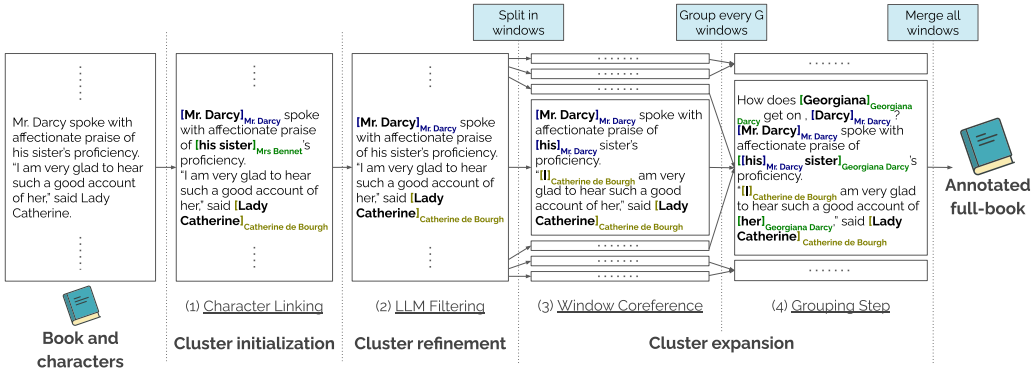


Figure 4.2. The BookCoref Pipeline applied on a sample taken from *Pride and Prejudice*. (1) Link all explicit character mentions via Character Linking. (2) Filter out inconsistent assignments via LLM Filtering. (3) Expand character clusters using a CR model on small windows. (4) Expand character clusters of grouped windows.

4.3.1 Cluster Initialization

In our first step, we initialize character coreference clusters[6] by tagging explicit entity mentions (i.e., non-pronoun mentions such as proper nouns or noun phrases). We extract these mentions through Entity Linking (EL), the task of linking explicit mentions of entities in a text to their corresponding entry in a Knowledge Base (KB). In our case, the index (or KB) of possible entities changes for each book b and corresponds to the list of character names $\mathcal{C}(b) = \{c_1, c_2, \ldots, c_n\}$. Formally, given a character c from a book $b \in \mathcal{B}$, we initialize its coreference cluster $\mathcal{E}_c^b$ with

[6]By character coreference cluster, we refer to the set of coreferential mentions that correspond to the same character.

all its explicit mentions extracted through an automatic EL system. We note as $\left\{\mathcal{E}_c^b\right\}_{c\in\mathcal{C}(b)} = \left\{\mathcal{E}_{c_1}^b, \mathcal{E}_{c_2}^b, \ldots, \mathcal{E}_{c_n}^b\right\}$ the set of character coreference clusters in a book b.

The main limitation of current pre-trained EL systems is that they are trained to link named entities to predefined knowledge bases, such as Wikipedia or Wikidata; in our case, instead, we aim to link explicit mentions of characters to their respective names, a task that we define as Character Linking. To this end, we prepare a small training set for the Character Linking task. Specifically, we manually amend LitBank (Bamman et al., 2020) by filtering out all non-explicit character mentions and naming each cluster with character names. We then use this resource to fine-tune a state-of-the-art EL system (Orlando et al., 2024, ReLiK) for Character Linking in the narrative genre.

We apply this trained system to our collection of books $\mathcal{B}$, obtaining the aforementioned character coreference clusters $\left\{\mathcal{E}_c^b\right\}_{c\in\mathcal{C}(b)}$, for each book $b \in \mathcal{B}$. As a by-product, our clusters are linked to a specific name, which is crucial to the next steps of our pipeline. We now reserve a brief section for the description of the technique used to train our character linking model on LitBank.

Character Linking on LitBank

Our main aim is to create a dataset that can be used to fine-tune ReLiK (Orlando et al., 2024) to perform Character Linking on the narrative text. To accomplish this, we adapt the LitBank Coreference Resolution dataset based on the following intuition: if we are able to assign a character name to each co-referring cluster of mentions associated with a character, we can then re-frame the task as linking each specific mention to its character name. More practically, the coreference data in LitBank comes with a categorization of entities across six ACE 2005[7] categories (people, facilities, locations, geo-political entities, organizations, and vehicles); we therefore only keep coreference clusters where all mentions are of type *people*, including both named entities and common nouns. Moreover, for each mention in a cluster, we are given its manually-validated part-of-speech tag. We filter out any pronoun mentions from all the clusters, as they are not usually tagged in Entity Linking settings.

[7] `https://catalog.ldc.upenn.edu/LDC2006T06`

We then manually name each remaining cluster, taking inspiration from the most frequently occurring proper nouns in a cluster but also taking clues from the book (e.g., naming a cluster "Tom Sawyard" even if the most frequently occurring PROP is "Tom"). This leaves us with a complete Character Linking dataset, i.e., an Entity Linking dataset where all the entities are anthropomorphized characters. The index of possible entities is the set of named coreference clusters of a book and is different for each book.

We then follow the instructions in the ReLiK repository[8] to fine-tune the ReLiK Reader on the LitBank training set. Evaluating on the held-out testing set returned an F1 score of 83.1.

4.3.2 Cluster Refinement

We place a strong emphasis on the precision of our initial cluster creation: false positives (i.e., incorrect mention-character links predicted by our system) can easily propagate through the whole pipeline, whilst false negatives (i.e., correct mention-character links that are not predicted by our system) are easier to recover in our subsequent Cluster Expansion step. Therefore, we introduce an additional verification step for each mention of a character cluster by prompting an LLM to ascertain whether the mention is accurately linked to the character, based on the surrounding context.

Specifically, we prompt Qwen2 7B Instruct (Yang et al., 2024)[9] with the name of the character and the highlighted mention in context, and we constrain the LLM to output either 'Yes' or 'No' as the answer to our prompt (see Table 4.2 for the exact prompt).

We filter out mentions for which the LLM answers negatively. The final result of this step is a set of LLM-validated coreferential clusters $\left\{\hat{\mathcal{E}}_c^b\right\}_{c\in\mathcal{C}(b)}$ for each character c of all our books $\mathcal{B}$.

[8] `https://github.com/SapienzaNLP/relik`

[9] We chose Qwen2 7B due to its permissive Apache 2.0 license.

Prompt
I will give you an excerpt from a book with a highlighted mention of a character with []. You will need to answer if the assigned character is correct (Yes), or not (No). Book excerpt: $book Does the mention [$mention] correspond to the character $character? (Yes/No)

Table 4.2. Prompt template for the Cluster Refinement step. The dollar sign ($) indicates a template variable.

4.3.3 Cluster Expansion

As a result of the previous steps, for each book, we have a set of high-precision annotations of coreference clusters containing the explicit mentions of each character that appears in that particular book.

To fully tag our books with complete CR annotations, we expand each coreference cluster by including every missing mention of a given character, including pronouns, noun phrases, and other references not tagged in the previous steps. We use Maverick because it combines state-of-the-art performance with the ability to complete partial coreferential clusters, which is a prerequisite in order to exploit the annotations obtained in the previous steps. However, the problem with using Maverick in a long-context setting is that it suffers from the same input length issues that impact other CR models, i.e., a huge memory cost that is implied by its full-attention mechanism. Therefore, for each book $b \in \mathcal{B}$, we apply Maverick to consecutive, non-overlapping windows $\mathcal{W}(b) = \{w_1, \ldots, w_N\}$, with each window w_i containing a maximum number of 1500 words, the closest setting to Maverick's training conditions. More precisely, for each window $w_i \in \mathcal{W}(b)$ in book b, we derive from the set of coreferential clusters produced in the previous steps $\left\{\hat{\mathcal{E}}_c^b\right\}_{c \in \mathcal{C}(b)}$ a local set of clusters $\left\{\hat{\mathcal{E}}_c^{w_i}\right\}_{c \in \mathcal{C}(b)}$ by filtering out all mentions of a cluster that are not contained in window w_i. Following this, we use Maverick on window w_i to expand each character cluster $\hat{\mathcal{E}}_c^{w_i}$ to all of its coreferential mentions, including pronouns, noun phrases and other references, obtaining a local and complete set of clusters $\{\mathcal{M}_c^{w_i}\}_{c \in \mathcal{C}(b)}$ for a window w_i of a book b.

To obtain a full-book annotation for b, we merge the annotations from all the

windows $w_i \in \mathcal{W}(b)$. This can be done by taking the union of every cluster linked to the same character c across all windows: $\mathcal{M}_c^b = \bigcup_{i=1}^{N} \mathcal{M}_c^{w_i}$. We leverage the fact that local coreference clusters are each linked to the same global set of character names, as discussed in the previous section.

Although merging all windows $w_i \in \mathcal{W}(b)$ results in a precise annotation of book b, this approach does present a particular edge case that affects the recall of character mentions: when a window w_i lacks explicit mentions of a character $c \in \mathcal{C}(b)$, our Cluster Expansion step becomes ineffective, as there are no mentions from which to expand and form the corresponding coreferential cluster. This is outlined in Figure 4.2, where the mention "his sister" is not linked to the character *Georgiana Darcy*.

To mitigate this limitation, we propose an intermediate grouping step in which we merge G consecutive windows into a single grouped window[10], and run a second expansion step on a context that is larger and contains more characters. We therefore introduce Maverick$_{xl}$ an adaptation of Maverick to processes longer documents. Practically, we adapt Maverick's architecture at inference time, calling it Maverick$_{xl}$, by encoding input documents in smaller sections (cf. Section 4.6.2, and use it to perform this second cluster expansion step. Formally, we define non-overlapping, grouped windows $gw_j \in \{gw_1, \ldots, gw_M\}$, with $M = \lceil |\mathcal{W}(b)| \,/\, G \rceil$, as the union of G consecutive windows, $gw_j = \bigcup_{i=G\cdot(j-1)+1}^{\min(G\cdot j,N)} w_i$, where $w_i \in \mathcal{W}(b)$. For each gw_j, we define its set of character clusters as the union of the local coreference clusters of the grouped windows: $\left\{\mathcal{M}_c^{gw_j}\right\}_{c\in\mathcal{C}(b)} = \left\{\bigcup_{i=G\cdot(j-1)+1}^{\min(G\cdot j,N)} \mathcal{M}_c^{w_i}\right\}_{c\in\mathcal{C}(b)}$. This grouping step expands the character clusters by considering the broader context provided by the grouped windows. We apply Maverick$_{xl}$ to each grouped window gw_j starting from $\left\{\mathcal{M}_c^{gw_j}\right\}_{c\in\mathcal{C}(b)}$ and we obtain a local set of coreference resolution annotations $\left\{\mathcal{F}_c^{gw_j}\right\}_{c\in C(b)}$. The benefits of this step can be seen clearly in Figure 4.2, where the mention "his sister" is correctly linked to the character *Georgiana Darcy*. Finally, for each character cluster, we take the union across these grouped windows gw_j, obtaining coreference annotations $\left\{\mathcal{F}_c^{b}\right\}_{c\in C(b)}$ for each book, which altogether represent the final coreference annotation of BookCoref$_{SILVER}$.

[10]By performing a statistical analysis on our corpus, we found that $G = 10$ results in an optimal intermediate length between the size of windows and full books.

Datasets	Docs	Tok.	Ment.	Tok./Doc	Ment./Doc	Chains/Doc	Ment./Chain	Ment. Dist.
OntoNotes	3,493	1.6M	194k	467	55	12.7	4.3	140
LongtoNotes	2,415	1.6M	194k	674	80	16.7	4.9	371
LitBank	100	210k	29k	2,105	291	79.3	4.1	480
WikiCoref	30	59k	7k	1,996	233	49.4	4.5	1,013
MovieCoref	9	201k	26k	22,423	2,865	46.4	89	7,672
BookCoref$_{GOLD}$	3	229k	24k	76,419	7,844	22.3	359	34,880
BookCoref$_{SILVER}$	50	10.8M	968k	216,626	19,471	27.4	870	73,432

Table 4.3. Statistics of BookCoref compared with previous CR datasets. Columns from left to right: total number of documents, tokens, and mentions; average number of tokens, mentions, and coreference chains per document; micro-average number of mentions per chain; and micro-average pairwise distance between mentions of the same chain. We use (k) and (M) to indicate thousands and millions, respectively.

4.4 BookCoref silver

4.4.1 Statistics

In Table 4.3 we report the statistics of our two new resources, BookCoref$_{SILVER}$ and BookCoref$_{GOLD}$, compared to well-established long-document benchmarks. Some distinctly book-scale characteristics emerge, such as very long documents, a low number of highly populated chains for each document, and a high average pairwise distance between mentions of the same chain.[11] BookCoref$_{SILVER}$ is both large-scale, with 6.75 times the number of tagged tokens and 5 times the amount of mentions compared to OntoNotes, and book-level, with mentions linked across unprecedented distances. Despite containing only three manually-annotated books, we note that BookCoref$_{GOLD}$ contains 229k tokens, ~10 times more than the test splits of well-established long-document benchmarks, LitBank (~21k) and MovieCoref (~29k).

4.4.2 BookCoref Pipeline Evaluation

We measure the performance of each step of our BookCoref Pipeline on our manually curated subset of books, BookCoref$_{GOLD}$. Table 4.4 reports traditional CR metrics and standard metrics for Character Linking. In particular, we measure MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), CEAF$_{\phi_4}$ (Luo, 2005) and their average value, CoNLL-F1, for Coreference Resolution, and precision, recall

[11]The average pairwise distance is calculated as the micro-average of all the distances, measured in number of tokens, between any two mentions that are part of the same chain.

	Character Linking			Coreference			
	P	R	F1	MUC	B^3	$CEAF_{\phi_4}$	CoNLL
Cluster Initialization							
Pattern matching	95.6	18.9	29.2	14.7	4.7	34.5	17.9
Character Linking (CL)	88.6	30.9	44.5	39.3	12.7	50.5	34.2
Cluster Refinement							
CL + LLM filtering	93.8	29.8	43.5	37.5	50.9	12.9	33.9
Cluster Expansion							
CL + Window Coreference	85.7	75.3	80.2	85.7	62.9	65.3	71.3
+ Grouping step	78.7	87.8	83.0	91.0	65.2	67.8	74.7
CL + LLM filtering + Window Coreference	90.3	80.4	84.7	85.6	67.0	78.6	77.7
+ Grouping step	83.2	89.8	86.3	93.3	70.8	77.5	80.5

Table 4.4. Evaluation of our BookCoref Pipeline on BookCoref$_{GOLD}$. We report precision, recall and F1-score for Character Linking and measure MUC, B^3, $CEAF_{\phi_4}$ and CoNLL-F1 as coreference metrics.

and F1-score for Character Linking. We evaluate our initial Cluster Initialization step performance, comparing our approach to a simple Pattern Matching (PM) baseline, where character clusters are initialized by selecting mentions that match the character name. Our Character Linking method obtains high precision scores and ensures a higher recall of character mentions compared to PM, resulting in an increase of +15.3 in F1-score. Applying LLM-based filtering on top of our Character Linking outputs increases the character precision by an average of +5.2 points, as it removes several wrongly-predicted mentions. Keeping these mentions would result in a propagation of annotation errors in the following Cluster Expansion step: in Table 4.8, we specifically present a qualitative analysis for this step. Regarding the final Cluster Expansion step, we show that the intermediate grouping strategy improves coreference-specific metrics, reaching 80.5 CoNLL-F1 score, and therefore we adopt it in the final BookCoref Pipeline. Finally, we highlight that our silver-annotation pipeline achieves an MUC score of 93.3, comparable to the inter-annotator MUC agreement between our human annotators (96.1) and between the annotators of other datasets (LitBank 95.5, OntoNotes 83.0, cf. Section 4.5).

4.5 BookCoref gold

To enable a complete and rigorous evaluation of coreference resolution models at book scale, we put forward a manually annotated benchmark, BookCoref$_{GOLD}$. We include the annotation provided by Guo et al. (2024) for *Animal Farm* by George

Orwell and two new, fully-annotated books, i.e., Herman Hesse's *Siddhartha* and Jane Austen's *Pride and Prejudice*. This latter is a particularly challenging benchmark as previous work has shown that its high density of characters and pronominal expressions often leads CR systems to produce erroneous links (Vala et al., 2015). Each annotator is presented with the full text of the book and the corresponding list of characters, and is tasked with tagging any mention of the characters by means of an annotation tool.[12] The annotation was conducted by three expert CR annotators for around 120 hours, covering a total of 194,280 words. Following recent literature, the agreement was computed on a 2,000-word subset using traditional coreference metrics, achieving a 96.1 inter-annotator MUC score – comparable to LitBank (95.5) and higher than OntoNotes (83.0). Notably, restricting annotations to characters contributes to a high agreement.

Annotation Guidelines As in the annotation of *Animal Farm* (Guo et al., 2024), we provide the annotators with a list of characters appearing in each book, and they are tasked with extracting co-referring mentions of each character. We follow previous works (LitBank, OntoNotes) in selecting as co-referring mentions spans of text that are part of one of the following categories: proper nouns (for example, *Siddhartha*); common nouns (*the gentleman*); personal pronouns (*he*, *she*); possessive pronouns (*his*, *hers*).

We also decide to mark the maximal extent of a span, resulting in noun phrases such as *"one of the most eminent physicians"*, *"a man whom nobody cared anything about"*, or *"the Right Honourable Lady Catherine de Bourgh"*. This latter example showcases our approach to honorifics, which we include in the maximal span without annotating them separately.

Regarding singleton mentions (i.e., noun phrases that do not co-refer with other mentions), we allow them to be tagged by our annotators, but because the task pertains to identifying co-referring mentions of a list of characters for each book, there are no instances of singleton mentions in BookCoref$_{GOLD}$.

Finally, we also instruct annotators to avoid annotating mentions that can be linked to more than one antecedent mention (referred to as *split-antecedent*), in line

[12] `https://github.com/nilsreiter/CorefAnnotator`

with most current Coreference Resolution datasets.

4.6 Experimental setup

In this Section, we report the experimental setup we developed to empirically analyze the training and testing of current systems on BookCoref.

4.6.1 Experiments Design

We benchmark automatic coreference systems on the proposed BookCoref dataset. Specifically, we test the models both off-the-shelf, i.e., loading their pre-trained configurations, and after training them on BookCoref$_{SILVER}$. We measure CR performance in two specific settings: i) on BookCoref$_{GOLD}$, in which models are evaluated on book-scale coreference by taking in input full books; ii) on Split-BookCoref$_{gold}$, in which models are evaluated on medium-sized texts that result from splitting BookCoref$_{GOLD}$ into independent windows of 1500 tokens. The two settings allow us to evaluate the ability of current CR systems to overcome the intrinsic difficulty of processing book-scale texts in comparison with medium-sized ones.

4.6.2 Comparison Systems

As already anticipated in our background Section 2.3.2, many modeling approaches (LLMs, seq-to-seq, encoder-only) present significant limitations. We now list the limitations of many current approaches, which motivate their exclusion, and then introduce the models used in our experiments.

Excluded systems

Discriminative models Discriminative models formulate the CR task as a classification problem. They typically employ a two-step formulation, in which the model first learns to extract mentions and then predicts their coreference relations. Recent models use an underlying encoder-only pre-trained Transformer architecture to encode the input documents. This gives rise to two main limitations: i) pre-trained

Transformers usually have a fixed maximum input length, and ii) their attention mechanism involves a quadratic computational complexity with respect to the input text. These two characteristics strongly limit their applicability to the proposed book-scale setting, in which input books can reach more than 600.000 tokens.

Sequence-to-Sequence models Sequence-to-sequence models have recently proven particularly robust for resolving coreference relations in well-established CR benchmarks(Bohnet et al., 2023; Zhang et al., 2023). However, the limitations that make the usage of discriminative models difficult in the full-book setting are even more marked when considering Sequence-to-Sequence architectures. In fact, these models usually rely on very large pre-trained Transformers with several billion parameters, and many works report that their applicability is limited by their computational requirements (Zhang et al., 2023; Porada et al., 2024b). An additional limitation is inherently implied by their application to the coreference task, which requires re-generating the full input text with the addition of special tokens that denote coreference. In this regard, Bohnet et al. (2023) reports problems of hallucinations and ambiguous matches of mentions to the input when dealing with the documents in OntoNotes. For these reasons, we note that the proposed book-scale setting is particularly challenging for generative systems with current formulations, and more advancements are needed to adapt them to this extended-length scenario.

LLMs The limitations we raise for large generative Coreference Resolution systems are exacerbated when attempting to apply Large Language Models (LLMs) to this task. To the best of our knowledge, there is no clear consensus on how LLMs should be used to extract and resolve mentions of co-referring entities without the pre-identification of mentions or the appearance of hallucinated text. Previous work also shows that, on the task of Coreference Linking, LLMs do not surpass the performance of smaller, bespoke models on medium-scale settings (Le and Ritter, 2024; Porada and Cheung, 2024). Moreover, the scale of our benchmark would constrain us to use only LLMs that have a context window larger than twice the length of the longest document in BookCoref$_{GOLD}$, i.e., more than 300,000 tokens. This would exceed our budget both economically and computationally, as we cannot

afford to run such systems for those input lengths locally, nor afford the costs of running them externally via APIs.

Tested Systems

In our experiments, we benchmark available solutions for long documents and adapt a state-of-the-art coreference system to the book-scale setting. Among long-document systems, we benchmark the widely-adopted BookNLP library[13], a BERT-based model that is trained on LitBank. We also include Longdoc (Toshniwal et al., 2020, 2021) and Dual cache (Guo et al., 2024), two systems that were specifically designed to handle long texts through an incremental formulation of CR. Those two architectures adopt an incremental formulation demonstrating their superiority for the long-document setting, but have never been previously tested on large book-scale benchmarks such as BookCoref.

We compare the results of these systems with Maverick, which currently achieves state-of-the-art results on OntoNotes. To use Maverick on our book-scale setting, we adapt its architecture at inference time: instead of encoding documents in their entirety, we perform mention extraction on smaller windows of arbitrary length L. We then carry out the subsequent clustering steps on all the mentions of the books, drastically reducing the memory requirements that would be involved in encoding the whole book. We name this model Maverick$_{xl}$, and we set $L = 4000$, the largest value that allows us to run inference on BookCoref$_{GOLD}$ on our academic-budget setup, i.e., a single NVIDIA RTX-4090 with 24GB of VRAM. Notably, Maverick and Maverick$_{xl}$ are equivalent when testing the model on Split-BookCoref$_{gold}$, where the input text is shorter than 4000 tokens.

With the exception of BookNLP[14], we train our comparison systems on BookCoref$_{SILVER}$. We split documents and respective coreference clusters into separate windows of a maximum length of 1500 tokens. This length both complies with the maximum input length specified by the adopted Transformer encoders, and also enables us to train all the comparison systems on our hardware, avoiding the

[13] `https://github.com/booknlp/booknlp`

[14] BookNLP's codebase does not allow training on new datasets.

huge memory cost of training on full books[15].

4.7 Results

In Table 4.5, we benchmark current models on BookCoref$_{GOLD}$ and on Split-BookCoref$_{gold}$.

4.7.1 Performance of the Comparison Systems on BookCoref

	BookCoref$_{GOLD}$							Split-BookCoref$_{gold}$			
Models	Ani.	P.&P.	Siddh.	MUC	B^3	C_{ϕ_4}	CoNLL	MUC	B^3	C_{ϕ_4}	CoNLL
Off-the-shelf											
BookNLP	41.1	41.2	45.1	83.1	40.9	2.4	42.2	81.1	52.3	18.7	50.6
Longdoc	39.1	48.9	44.8	79.9	52.1	7.9	46.6	80.7	63.8	39.0	61.2
Dual cache	36.7	42.2	50.9	82.3	41.0	4.2	42.5	82.9	67.8	43.9	64.8
Maverick$_{xl}$	29.1	39.2	50.8	81.2	35.6	6.1	41.2	83.8	69.5	46.1	66.5
Fine-tuned on BookCoref$_{SILVER}$											
Longdoc	**67.5**	**63.9**	**74.0**	93.5	**62.4**	**45.3**	**67.0**	91.2	74.9	65.7	77.1
Dual cache	52.6	47.9	68.9	92.5	48.0	16.9	52.5	91.2	74.5	66.2	77.3
Maverick$_{xl}$	62.7	57.5	68.0	**94.3**	55.3	33.4	61.0	**92.7**	**82.2**	**71.9**	**82.2**

Table 4.5. Comparison between off-the-shelf models and systems trained on BookCoref$_{SILVER}$, tested on BookCoref$_{GOLD}$ and Split-BookCoref$_{gold}$. For each system, we report F1 measures of MUC, B^3 and CEAF$_{\phi_4}$ (noted as C_{ϕ_4}), and use their average, CoNLL-F1, as the main evaluation criteria. The first three columns detail avg. CoNLL-F1 scores on the three different books of our test-set, namely *Animal Farm*, *Pride and Prejudice*, and *Siddharta*. We highlight in **bold** the best measures of trained models, and underline best off-the-shelf results.

BookCoref$_{gold}$

In Table 4.5, we benchmark current models on BookCoref$_{GOLD}$ and on Split-BookCoref$_{gold}$. Off-the-shelf models all achieve a CoNLL-F1 score above 40 points, with Longdoc obtaining the best performance of 46.6. Notably, in this setting, the popular BookNLP performs similarly to other neural counterparts.

As expected, training our comparison systems on the BookCoref$_{SILVER}$ data consistently improves their performance. In this setting, Longdoc is still the best-performing model, scoring 67.0 CoNLL-F1 points (+20.4 compared to the off-the-shelf version), demonstrating the superiority of its incremental formulation. Interestingly, Dual cache, which was previously considered the best choice for long-document CR,

[15] For reference, training Maverick on a single book of 300k tokens requires more than 1200 GB of VRAM.

achieves low scores after fine-tuning, even when compared with Maverick$_{xl}$, a model that is not specifically tailored for long-document processing.

Our results demonstrate that none of the available off-the-shelf systems could have been considered a good candidate for the annotation of our silver corpus. In fact, the proposed BookCoref Pipeline obtains 80.5 CoNLL-F1 score on BookCoref$_{GOLD}$ (Table 4.4), +33.9 compared to the best off-the-shelf model.

Notably, by training on BookCoref$_{SILVER}$, Longdoc scores 67.5 CoNLL-F1 points on the Animal Farm benchmark, an increase of 31.2 points compared to previous literature (Guo et al., 2024). We reserve an additional experiments Section 4.7.2 for a more detailed report on the performance and robustness of the comparison systems.

Split-BookCoref$_{gold}$

Results on Split-BookCoref$_{gold}$ show an interesting finding: both off-the-shelf and fine-tuned systems perform much better in this medium-sized text setting, with Maverick$_{xl}$ reaching the best results. Specifically, Maverick$_{xl}$ scores 82.2 CoNLL-F1 points when trained on the BookCoref$_{SILVER}$ data, surpassing the second-best model, Dual cache, by +4.9 points.

We note that this result aligns with previous work, demonstrating that the incremental formulation of Longdoc is particularly robust on full books, whilst Maverick's single forward pass approach is superior on medium-sized text.

Efficiency

In Table 4.6, we detail the time and memory requirements of the comparison systems on our full book evaluation. Our results show that Longdoc achieves the lowest inference time and memory usage when processing BookCoref$_{GOLD}$. We also note that Longdoc improves its time efficiency after being trained on BookCoref$_{SILVER}$. We believe that the model learns to extract fewer, more precise mentions, which impacts its incremental clustering step and reduces processing time to 26 seconds.

	Time (sec.)	Memory (GB)
Off-the-shelf		
BookNLP	180	6.9
Longdoc	**70**	**5.8**
Dual cache	460	11.6
Maverick$_{xl}$	211	14.8
Trained on BookCoref$_{SILVER}$		
Longdoc	**26**	**5.8**
Dual cache	473	11.8
Maverick$_{xl}$	183	12.8

Table 4.6. Efficiency of each model when running inference on BookCoref$_{GOLD}$. We measure the total execution time in seconds and the maximum GPU memory in gigabytes, marking the best scores in bold. All experiments were run on an NVIDIA RTX-4090.

4.7.2 Metrics

Comparing the measures of the full-book setting with the ones obtained with smaller windows unveils a recurring pattern: traditional coreference scores, i.e., CEAF$_{\phi_4}$ B^3 and MUC, show discrepancies within each comparison system when evaluating full-book performance.

Interestingly, MUC often measures over 80 F1 points both in the window and book-scale setting, while B^3 and, in particular, CEAF$_{\phi_4}$, often show lower scores on full books, which substantially increase when evaluating model performance on smaller texts.

While many previous works have shown weaknesses and disagreements of these metrics in specific scenarios (Moosavi and Strube, 2016; Borovikova et al., 2022; Duron-Tejedor et al., 2023), we argue that this new proposed book-scale setting offers fresh ground for improving the study of robustness and expressivity of standard CR metrics. We now reserve an additional metrics analysis to obtain further insights into the behavior of coreference metrics and models in our book-scale scenario. Specifically, we propose an intermediate setting where full-book predictions are evaluated on smaller windows, and find that the discrepancy between CR metrics is lower in this setting compared to the original BookCoref$_{GOLD}$. These additional results highlight the need to increase the robustness of CR metrics in the book setting.

Additional Metrics Analysis

Setup We benchmark both off-the-shelf and trained systems on the two settings presented in Section 4.6 and propose a third setting to obtain further insights, BookCoref$_{GOLD+WINDOW}$. We now detail the three proposed settings:

- BookCoref$_{GOLD}$, the full-book setting. Models take in input full books and predict book-level coreference clusters that are scored against book-level gold clusters.
- Split-BookCoref$_{gold}$, the split-book setting: Models take in input books split into independent windows of 1500 tokens and produce window-level clusters, which are scored against the respective window-level gold clusters.
- BookCoref$_{GOLD+WINDOW}$, an intermediate setting, in which full-book predictions are evaluated in windows. Specifically, models take in input full books and predict book-level coreference clusters, as in the full-book setting. Then, performance is computed by considering only the predictions that refer to the specific windows used in Split-BookCoref$_{gold}$, scoring them against the respective window-level gold clusters.

Introducing our third intermediate setting is crucial for our investigation into the behavior of coreference metrics and models in our book-scale scenario.

Comparing the results between BookCoref$_{GOLD}$ and BookCoref$_{GOLD+WINDOW}$ is useful to evaluate whether the same predictions are underestimated by traditional coreference metrics. On the other hand, comparing the outcome of Split-BookCoref$_{gold}$ and BookCoref$_{GOLD+WINDOW}$ is useful to investigate changes in performances when having the full book context. As discussed in Section 4.6.2, many available coreference systems cannot be applied to the book-scale setting. We now reserve a subsection in which we detail current limitations, and then introduce our comparison systems.

Performance of comparison systems In Table 4.7 we report the results of our comparison systems in the three proposed settings. Interestingly, results on BookCoref$_{GOLD+WINDOW}$ are consistently higher compared to the ones measured

	BookCoref$_{GOLD}$				BookCoref$_{GOLD+WINDOW}$				Split-BookCoref$_{gold}$			
	MUC	B^3	CEAF$_{\phi_4}$	CoNLL	MUC	B^3	CEAF$_{\phi_4}$	CoNLL	MUC	B^3	CEAF$_{\phi_4}$	CoNLL
Off-the-shelf												
BookNLP	<u>83.1</u>	40.9	2.4	42.2	81.3	54.3	23.4	53.0	81.1	52.3	18.7	50.6
Longdoc	79.9	<u>52.1</u>	<u>7.9</u>	<u>46.6</u>	80.3	64.2	34.2	59.4	80.7	63.8	39.0	61.2
Dual cache	82.3	41.0	4.2	42.5	<u>82.6</u>	<u>67.4</u>	<u>41.3</u>	<u>63.7</u>	82.9	67.8	43.9	64.8
Maverick$_{xl}$	81.2	35.6	6.1	41.2	80.9	55.9	26.2	54.3	<u>83.8</u>	<u>69.5</u>	<u>46.1</u>	<u>66.5</u>
Fine-tuned on BookCoref$_{SILVER}$												
Longdoc	93.5	**62.4**	**45.3**	**67.0**	80.8	**75.3**	**62.2**	**76.2**	91.2	74.9	65.7	77.1
Dual cache	92.5	48.0	16.9	52.5	90.7	73.5	61.4	75.2	91.2	74.5	66.2	77.3
Maverick$_{xl}$	**94.3**	55.3	33.4	61.0	**91.7**	72.8	55.6	73.4	**92.7**	**82.2**	**71.9**	**82.2**

Table 4.7. Comparison between off-the-shelf models and systems trained on BookCoref$_{SILVER}$, tested on BookCoref$_{GOLD}$ Split-BookCoref$_{gold}$ and BookCoref$_{GOLD+WINDOW}$. For each system, we report F1 measures of MUC, B^3 and CEAF$_{\phi_4}$, and use their average, CoNLL-F1, as the main evaluation criteria. We highlight in **bold** the best measures of trained models, and <u>underline</u> best off-the-shelf results.

in BookCoref$_{GOLD}$, indicating that traditional coreference metrics are biased towards lower performances when evaluating full-book predictions. Furthermore, when evaluated on BookCoref$_{GOLD+WINDOW}$, the score increase is particularly high for Dual-cache, suggesting that this model is locally accurate but lacks full-book consistency. The same cannot be said for Longdoc, which, in the BookCoref$_{GOLD+WINDOW}$ setting, confirms the high performances obtained on full-book predictions. In general, all the systems perform better when taking in input medium-sized texts, as on Split-BookCoref$_{gold}$, suggesting that the current models are particularly tailored for processing shorter lengths.

Coreference metrics In the full book setting, we notice a low agreement between the MUC and CEAF$_{\phi_4}$ metrics. In particular, MUC scores are typically very high, whilst CEAF$_{\phi_4}$ instead assigns a low evaluation to the same predictions. This gap between these two metrics is less evident on BookCoref$_{GOLD+WINDOW}$, in which the same predictions are evaluated in smaller windows. Since analyzing this phenomenon is not in the scope of this thesis, we leave this research direction to future work. Nevertheless, we believe that a more detailed study is needed for analyzing metric behavior in full-book settings, and more attention should be devoted to the interpretation of their scores.

4.7.3 Error Analysis

We now put qualitative error analysis of our pipeline in Table 4.8. The first example demonstrates effective LLM Filtering, which correctly removes the mention "[her brother]$_{\text{Mr. Darcy}}$" that should link to 'Charles Bingley'. Without this filtering, Cluster Expansion would propagate the error by including nearby pronouns. When applied after LLM Filtering, Cluster Expansion successfully identifies the correct character by linking coreferential mentions across a wider context. The second example illustrates the pipeline's robustness even when LLM Filtering over-corrects by removing valid mentions like "[Mr. Darcy]$_{\text{Mr. Darcy}}$". The Cluster Expansion step recovers these filtered mentions while also correcting incorrectly linked mentions from the initial clustering ("[the daughter]$_{\text{Charlotte Lucas}}$").

4.7.4 Open Challenges of Book-scale Coreference Resolution

Our results show that Longdoc achieves the best scores on book-scale coreference resolution after training on BookCoref$_{SILVER}$.

We point out that the score of 67.0 CoNLL-F1 points on BookCoref$_{GOLD}$ is relatively low compared to the score models attain on other CR benchmarks such as OntoNotes and LitBank, which is around 80 CoNLL-F1 points. Nevertheless, the results obtained on Split-BookCoref$_{gold}$ show that models can reach similar scores when dealing with smaller windows of text, attaining 82.2 CoNLL-F1 points. This demonstrates that the difference in performance between BookCoref$_{GOLD}$ and Split-BookCoref$_{gold}$ is not due solely to the lack of manually annotated training resources. On the contrary, we argue that this difference highlights the open research challenges of our new book-scale setting, such as i) studying the interpretation of current coreference metrics for full-book evaluation, ii) creating new systems that can fully exploit book annotations, and iii) exploring efficient solutions for adapting current generative and encoder-only models for book-scale processing without incurring exponential computational costs.

Stage	Annotation
Character Linking (CL)	[Miss Bingley]$_{\text{Caroline Bingley}}$ sees that [her brother]$_{\text{Mr. Darcy}}$ is in love with you and wants him to marry [Miss Darcy]$_{\text{Georgiana Darcy}}$.
CL + LLM Filtering	[Miss Bingley]$_{\text{Caroline Bingley}}$ sees that her brother is in love with you and wants him to marry [Miss Darcy]$_{\text{Georgiana Darcy}}$.
CL + Window Coreference	[Miss Bingley]$_{\text{Caroline Bingley}}$ sees that [[her]$_{\text{Caroline Bingley}}$ brother]$_{\text{Mr. Darcy}}$ is in love with [you]$_{\text{Jane Bennet}}$ and wants [him]$_{\text{Mr. Darcy}}$ to marry [Miss Darcy]$_{\text{Georgiana Darcy}}$.
CL + LLM Filtering + Window Coreference	[Miss Bingley]$_{\text{Caroline Bingley}}$ sees that [[her]$_{\text{Caroline Bingley}}$ brother]$_{\text{Charles Bingley}}$ is in love with [you]$_{\text{Jane Bennet}}$ and wants [him]$_{\text{Charles Bingley}}$ to marry [Miss Darcy]$_{\text{Georgiana Darcy}}$.
CL	When, after examining [the mother]$_{\text{Lady Catherine de Bourgh}}$, in whose countenance and deportment she soon found some resemblance of [Mr. Darcy]$_{\text{Mr. Darcy}}$, she turned her eyes on [the daughter]$_{\text{Charlotte Lucas}}$, she could almost have joined in [Maria]$_{\text{Maria Lucas}}$'s astonishment at her being so thin and so small.
CL + LLM Filtering	When, after examining the mother, in whose countenance and deportment she soon found some resemblance of Mr. Darcy, she turned her eyes on the daughter, she could almost have joined in [Maria]$_{\text{Maria Lucas}}$'s astonishment at her being so thin and so small.
CL + Window Coreference	When, after examining [the mother]$_{\text{Lady Catherine de Bourgh}}$, in whose countenance and deportment [she]$_{\text{Elizabeth Bennet}}$ soon found some resemblance of [Mr. Darcy]$_{\text{Mr. Darcy}}$, [she]$_{\text{Elizabeth Bennet}}$ turned [her]$_{\text{Elizabeth Bennet}}$ eyes on [the daughter]$_{\text{Charlotte Lucas}}$, [she]$_{\text{Elizabeth Bennet}}$ could almost have joined in [Maria]$_{\text{Maria Lucas}}$'s astonishment at [her]$_{\text{Charlotte Lucas}}$ being so thin and so small.
CL + LLM Filtering + Window Coreference	When, after examining [the mother]$_{\text{Lady Catherine de Bourgh}}$, in whose countenance and deportment [she]$_{\text{Elizabeth Bennet}}$ soon found some resemblance of [Mr. Darcy]$_{\text{Mr. Darcy}}$, [she]$_{\text{Elizabeth Bennet}}$ turned [her]$_{\text{Elizabeth Bennet}}$ eyes on [the daughter]$_{\text{Miss de Bourgh}}$, [she]$_{\text{Elizabeth Bennet}}$ could almost have joined in [Maria]$_{\text{Maria Lucas}}$'s astonishment at [her]$_{\text{Miss de Bourgh}}$ being so thin and so small.

Table 4.8. Error analysis of our BookCoref Pipeline predictions on *Pride and Prejudice* from our test set. The table examines the three main pipeline steps as defined in Table 4.4 (Cluster Initialization, Cluster Refinement, and Cluster Expansion), through two examples.).

4.8 Summary

In this chapter, we fill the current gap in book-scale coreference resolution by proposing BookCoref, a novel benchmark with unprecedented book-scale characteristics. We put forward BookCoref$_{SILVER}$, a silver training corpus, and BookCoref$_{GOLD}$, a manually annotated dataset for model evaluation. BookCoref$_{SILVER}$ is annotated using the BookCoref Pipeline, a novel procedure that can provide full-text

coreference annotations of the characters in a book, starting from the character names. We carefully validate our automatic approach intrinsically, measuring its effectiveness on our manually annotated dataset, and also extrinsically, showing that long-document systems benefit from training on our silver resource. Nevertheless, our results on BookCoref$_{GOLD}$ show that specifically tailored long-document systems have a notable decrease in performance when evaluated on full books compared to that on smaller texts. By releasing BookCoref, we enable the investigation of the new research challenges posed by the book-scale setting, such as developing more accurate systems and studying long-document metrics. However, long documents are not the only challenging setting for current coreference models: Cross-document coreference, the task of solving clustering mentions across distinct documents, is even more challenging, with systems reaching only 35 CoNLL-F1 points on the main benchmarks. To address both challenges, in the next chapter, we put forward a novel approach that is capable of solving both cross- and long-document coreference, improving performances across multiple benchmarks.

~ This page was intentionally left blank ~

Chapter 5

Unifying Cross- and Long-document Coreference Resolution

Abstract

In the previous chapter, we detailed BookCoref, a new benchmark of fully annotated books that extends current resources up to 200,000 tokens per document. Moreover, we presented how current coreference resolution systems, tailored for short- or medium-sized texts, struggle to scale to very long documents due to architectural limitations and implied memory costs, and 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 chapter, 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. xCoRe 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.

Source: This chapter is based on the EMNLP 2025 paper: xCoRe: Cross-context Coreference Resolution.
Code: `http://github.com/sapienzanlp/xcore`.

5.1 Overview

As discussed in the previous chapter, while modern neural models have achieved 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 for very long documents, where models must maintain coherence over extended spans of text. A similar challenge arises in the cross-document setting, which requires reasoning over large inputs to link entities mentioned across multiple documents.

The transition from short to large contexts is non-trivial: most existing coreference techniques struggle with extended inputs due to the quadratic complexity of their Transformer-based architectures. To address this problem, recent solutions have proposed segmenting 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 work, 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 5.1, 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-

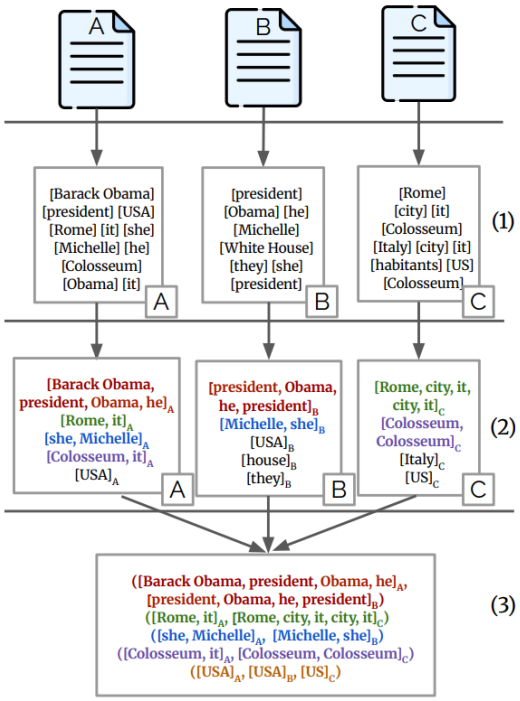


Figure 5.1. The xCoRe pipeline: for each input context, we adopt: (1) within-context mention extraction, to extract possible mentions, (2) within-context mention clustering, to build local clusters, and (3) cross-context cluster merging, to obtain the final set of cross-context clusters.

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.

5.2 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, i.e., mention

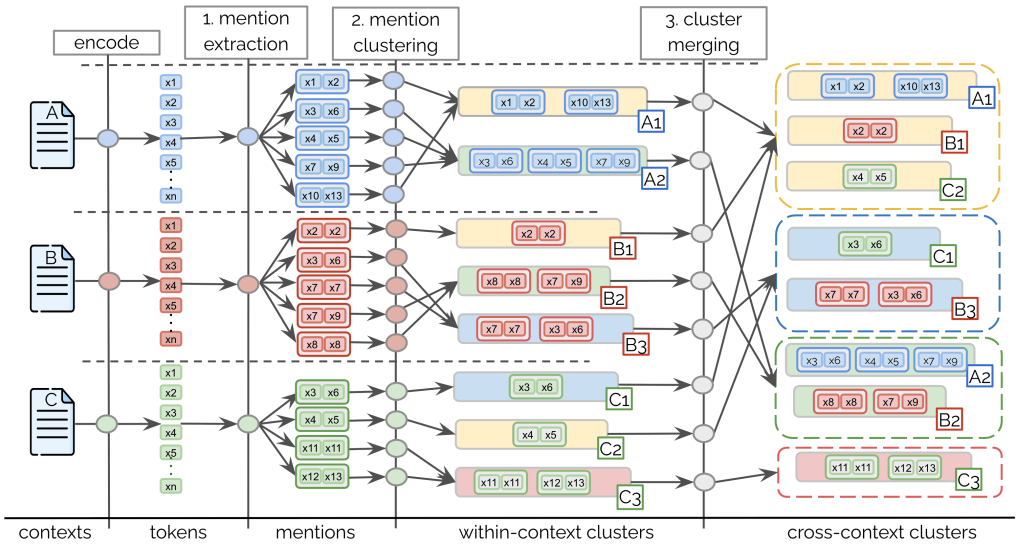


Figure 5.2. Illustration of the xCoRe architecture, which takes as input multiple contexts, illustrated as "A", "B", and "C", and outputs their merged coreference clusters. For each context, within-context clusters are extracted via within-context (1) mention extraction and (2) mention clustering. Finally, the cross-context (3) cluster merging step is applied to form clusters at the cross-context level.

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.

5.3 xCoRe Architecture

We now introduce our model pipeline, detailed in Figure 5.2. 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.

[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.

5.3.1 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 Maverick and serves as a stepping stone for our new cluster merging strategy, we provide a short overview of our within-context methodology here, and refer to Section 3.2 for the detailed discussion.

Mention Extraction

To extract mentions from each context $c_i \in C$, we adopt an equivalent approach to Maverick, presented in Section 3.2.1. 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 possible 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, presented in Section 3.2.2. 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 5.2, 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.

5.3.2 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 :

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

By taking the CLS, we can encode the variable-length set of mentions into a fixed-length output representing the within-document cluster. 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, and $\|$ specifies the concatenation of the two hidden representations. 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.

5.3.3 Cross-context Training and Inference

At inference time, as reported in Section 5.2, 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, 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.

Training Details

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. While the first two steps are inherently similar to the ones presented in Section 3.2.3, we also detail our third step, which is indeed a novel contribution of xCoRe:

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

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 5.3.1. Therefore, given all contexts $c_i \in B$, where B is the training batch prepared with our training strategy detailed in Section 5.3.3, i.e., we compute the mention extraction loss L_{extr} as:

$$\begin{aligned} L_{\text{start}}(c_i) &= \sum_{j=1}^{N} \ell_{\text{BCE}}(y_j,\, p_{\text{start}}(t_j)) \\ L_{\text{end}}(c_i) &= \sum_{s=1}^{S} \sum_{k=1}^{E_s} \ell_{\text{BCE}}(y_{sk},\, p_{\text{end}}(t_k | t_s)) \\ L_{\text{extr}} &= \sum_{i=1}^{|B|} L_{\text{start}}(c_i) + L_{\text{end}}(c_i) \end{aligned}$$

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 mentions 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 the Maverick chapter, Section 3.2.2.

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 5.3.2. 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

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 5.1. 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.

merging step. This loss guides the final step of the pipeline by training the model to correctly predict whether two clusters from separate contexts $\mathcal{W}_a^{c_i}$ and $\mathcal{W}_b^{c_j}$ refer to the same entity.

5.4 Experimental Setup

We evaluate xCoRe capabilities on well-established standard, long- and cross-document coreference datasets. Moreover, in our experiments, we report extensive ablation studies to validate the effectiveness of our cross-context cluster merging strategy and the impact of using gold information for mention extraction clustering and merging. We now discuss datasets and comparison systems in Section 5.4.1 and Section 5.4.2, respectively. Finally, we introduce xCoRe models, baselines, and setups in Section 5.4.3

5.4.1 Datasets

We now report technical details of the benchmarks adopted in the following sections, and refer the reader to Table 5.1 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* – 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. 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. On LitBank, our comparison systems are trained and tested with singleton mentions on LB_0, the first cross-validation fold of LitBank. Finally, we also include results on traditional benchmarks such as OntoNotes (Pradhan et al., 2012) and PreCo (Chen et al., 2018).

5.4.2 Comparison Systems

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

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+ (the "Cybulska setting") and has never been tested on SciCo, we replicate its results to be consistent with recent techniques (i.e., removing access to the additional context, resulting in lower performance) and note it as PMCoref†. 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 (AI, 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; however, since they do not include the weights in the original repository, we adopt a recent implementation of the Hierarchical model[2]. We also include Maverick, which adopts the traditional mention-to-antecedent scoring strategy. Additionally, 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.

5.4.3 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 5.4.1).

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

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

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.

Setup In our experiments, Maverick systems adopt DeBERTA−v3 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 on Maverick experiments as detailed in Section 3.4.4. All our experiments are run on an academic budget, i.e., a single NVIDIA RTX-4090.

5.5 Results

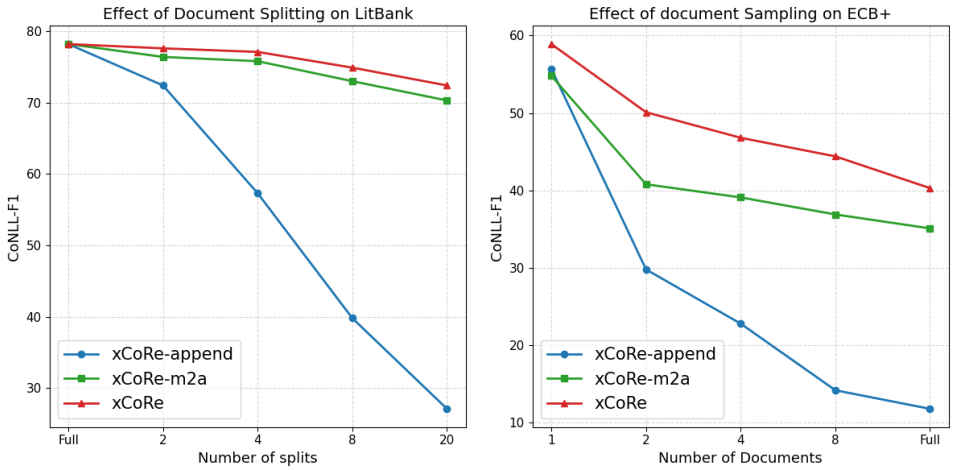
5.5.1 Cluster Merging Analysis

We first analyze the impact of the cluster merging approach, and report our results in Table 5.2 and in Figures 5.3a and 5.3b. Specifically, we evaluate xCoRe, xCoRe-*append*, and xCoRe-*m2a* on *LitBank-Split*, 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,

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 5.2. 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.



(a) CoNLL-F1 scores comparison on LitBank with increasing number of splits per document.

(b) CoNLL-F1 scores comparison on ECB+ with increasing number of documents per topic.

confirming the superiority of our method based on merging locally extracted clusters.

5.5.2 Cross-document Benchmarks

In Table 5.3 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 addi-

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
xCoRe$_{\text{LitBank}}$	**42.4**	74.1	**30.5**	67.3

Table 5.3. 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.

tional 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).

5.5.3 Long-document Benchmarks

As outlined in Table 5.4, 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

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 5.4. 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.

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, 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).

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.

5.5.4 Medium-size Benchmarks

In Table 5.5, 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, 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.

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

Table 5.5. Step-wise error analysis of **xCoRe** across all datasets (CoNLL-F1 score). We show the base model, a variant with gold mentions, and one with both gold mentions and gold clusters.

5.5.5 Step-wise Error Analysis

To further analyze the effectiveness of our pipeline, in Table 5.5 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, since it only annotates book characters. 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 most challenging step of the pipeline is cluster merging. This result suggests that focusing on advancing our proposed simple yet effective clustering technique could lead to additional improvements in every coreference scenario.

5.6 Summary

In this chapter, 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-sized and long-document datasets. By releasing this model, we potentially benefit several downstream applications, filling the gap for an end-to-end, robust system across challenging coreference scenarios.

~ This page was intentionally left blank ~

Chapter 6

Semantically Enhanced Coreference Resolution Evaluation

Abstract

In the previous chapters, we focused on advancing methods and resources for coreference resolution, with particular attention to long and multi-document texts. An equally important yet often overlooked challenge, as discussed in Section 2.5.1, is the evaluation of coreference systems. The scarcity of neural-based evaluation metrics limits our ability to assess models in a way that is both comprehensive and interpretable. In this chapter, we propose leveraging coarse-grained semantic typing to enrich the evaluation of coreference predictions. Specifically, we introduce CNER, a task that assigns semantic categories to both named entities and abstract concepts (typically expressed as common nouns), and use it to infer the most probable semantic class of each coreference cluster. By adding this semantic perspective, we can diagnose whether a model underperforms in particular domains or struggles with specific entity types. We argue that this approach not only improves interpretability by providing meaningful semantic labels for clusters, but could also support the development of more robust coreference models: by identifying which semantic classes are systematically misclassified or underrepresented, we could better target domain-specific corpora during future trainings. In the remainder of this chapter, we first introduce the CNER task, a novel contribution of this thesis, then describe our methodology for mapping semantic classes to coreference clusters, and finally present our evaluation experiments and findings.

Source: Based on our NAACL 2024 paper, *CNER: Concept and Named Entity Recognition* (Martinelli et al., 2024b), as well as additional novel experiments.
Code: `https://github.com/babelscape/cner`

6.1 Overview

As outlined in Section 2.5, most current evaluation techniques for coreference are statistical methods and exhibit several limitations. They often focus on the presence or absence of correct mentions and links between mentions, yet they struggle to capture semantics or contextual nuances, leading to a lack of semantic understanding. For example, many widely used metrics require exact span matches, and therefore minor but semantically irrelevant boundary discrepancies are treated as complete errors, complicating comparisons across different annotations, genres, or languages.

Interpretability is another key issue: a single global score often obscures systematic failure modes. For example, it may be unclear whether errors arise from missing rare entities, mislinking pronouns, or fragmenting salient characters, therefore masking qualitatively different types of errors and making progress harder to interpret.

To address these challenges, we introduce CNER, a framework for categorizing both named entities (typically expressed as proper nouns) and concepts (often realized as common nouns) into the same unified set of semantic types. We then propose a simple overlap-based method for assigning each coreference cluster a dominant semantic class, thus enabling a semantically grounded evaluation of cluster quality. Our experiments show that this approach improves interpretability by highlighting which semantic categories exhibit low performance, while also providing actionable insights for model development: by identifying systematically misclassified or underrepresented classes, we could guide targeted, domain-specific improvements.

In the remainder of this chapter, we first present the Concept and Named Entity Recognition task, along with the associated resources and models. Then, in Section 6.3, we describe our methodology for expanding CNER annotations to coreference mentions. Finally, Section 6.4 details our experimental setup, and Section 6.5 discusses the results of our evaluation.

6.2 Concept and Named Entity Recognition

Coreference chains do include not only named entities (e.g., *London*, *Barack Obama*) but also a wide variety of common nouns and abstract concepts (e.g., *the city*, *the president*, *the policy*, *the device*). While Named Entity Recognition (NER) provides valuable semantic information for many proper nouns, it largely overlooks these nominal and conceptual mentions, even though they are pervasive in real-world texts and crucial for understanding coreference structure. For our goal of semantically enhanced coreference evaluation, relying solely on traditional NER leaves most mentions untyped and therefore limits our ability to analyze model errors by semantic category.

To bridge this gap, we introduce the Concept and Named Entity Recognition (CNER) task: a unified approach that assigns semantic types not only to named entities but also to nominal concepts. CNER extends NER with a broader and richer inventory of semantic classes while maintaining a single, coherent classification scheme for both. This unified approach allows us to semantically characterize the full set of nominal mentions in a document, providing a far more informative basis for diagnosing coreference model performance. As illustrated in Figure 6.1, this joint formulation produces denser, richer, and more cohesive semantic annotations than traditional named entity recognition alone.

In the following sections, we describe the CNER task, the semantic categories we define, and the resources and models we employ to annotate both entities and concepts consistently.

6.2.1 The CNER task

We start by formalizing the Concept and Named Entity Recognition task and then introduce our process to obtain CNER categories. Formally, CNER is a sequence labeling task whose goal, given a sequence of tokens $t = (t_1, t_2, ..., t_n)$, is to identify the spans of text S corresponding to nominal concepts and named entities, and categorize each of them into a predefined set of labels $L = \{l_1, l_2, ..., l_m\}$. Specifically, a text span $s_{ij} \in S$ is defined as a contiguous sequence of tokens $s_{ij} := (t_i, t_{i+1}, \ldots, t_j)$, with $1 \leq i \leq j \leq n$.

Figure 6.1. Four examples of CNER annotations where both nominal concepts and named entities are tagged with the proposed categorization.

6.2.2 CNER Categories

To design our comprehensive set of categories suitable for both concepts and named entities, we draw upon the robust cognitive foundations of WordNet (Miller, 1995) and OntoNotes (Pradhan et al., 2012). We note that, in this section, with OntoNotes, we refer to the NER annotations and categories presented in the CoNLL-2012 shared task. Specifically, we leverage the completeness and broad semantic coverage of lexicographer files for nominal concepts, and integrate them with the widely-used semantic categories for named entities available in OntoNotes. Figure 6.2 shows the resulting 29 CNER categories (inner column) together with their alignment with the OntoNotes and WordNet ones (left and right columns, respectively). Every category in OntoNotes and WordNet has a counterpart in our categorization, whereas the reverse does not hold. Specifically, i) every category highlighted in red has a direct link with a WordNet category (e.g. ASSET), ii) every category highlighted in yellow corresponds to an OntoNotes category (e.g. MONEY), and iii) every category in a white cell is grounded on both resources (e.g. STRUCT).

To further assess the validity of such categorization and ensure that each instance is assigned to a reasonable category (i.e. semantically appropriate and neither too general nor too specific), we conducted a manual review of the most prominent WordNet synsets, i.e. the top 500 synsets ranked by their number of descendants in the WordNet taxonomy: a group of three human annotators independently reviewed this set of synsets and either assigned a category that is present in the already available categorization or proposed a new one, i.e. a category whose semantics better defines such an instance and that is well distinguished from the others. The result of this step is twofold. First, we obtained 500 synsets tagged with their CNER category, that we refer to as *seed synsets*, which will be employed as a starting point for our automatic annotation procedure, explained in Section 6.2.4. Second, we identified six new categories, highlighted in green in Figure 6.2, that complement the WordNet and OntoNotes ones. The practical implication of the latter result is that, for example, biological concepts such as *molecule*, *protein*, and *organism* will no longer be tagged with the OBJECT category, but as BIOLOGY. Analogously, terms such as *planet*, *quasar*, and *star*, that would all have been

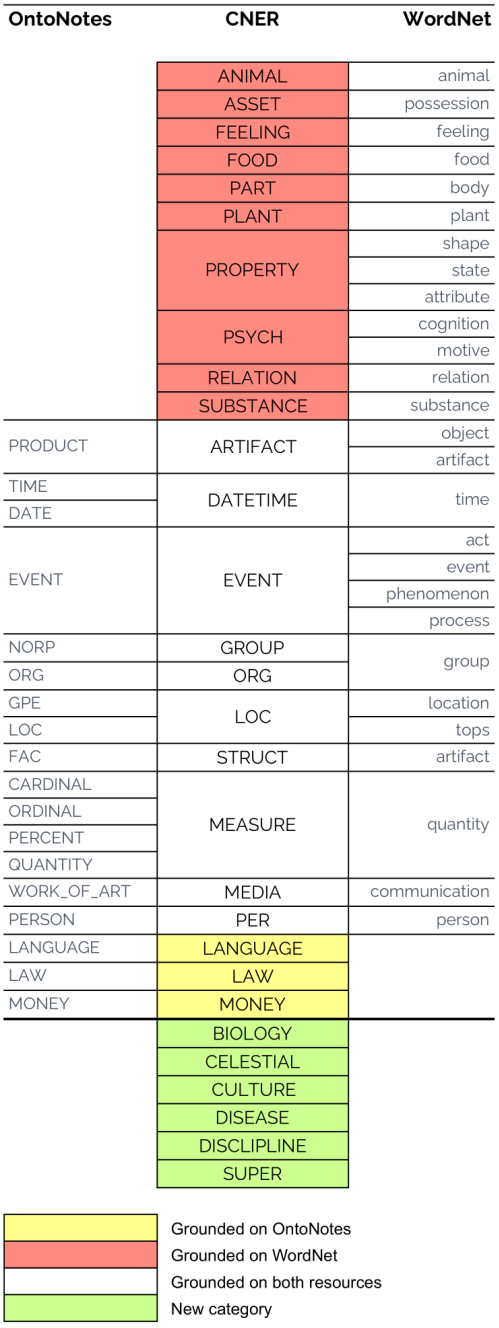


Figure 6.2. Comparison of our CNER categories (inner column) with the OntoNotes (left column) and WordNet ones (right column).

included in the Object category, will now be assigned to a specific category named Celestial. The same reasoning can be extended to the new categories Culture, Disease, Discipline, and Super. For completeness, in Table 6.1 we provide a textual description along with instance examples for each CNER category.

6.2.3 Resources

In the previous section, we described the procedure we used to obtain a new comprehensive categorization specifically designed to jointly capture concepts and named entities. We now provide datasets to enable the training and evaluation of CNER models. Crucially, we note that none of the existing datasets provides full coverage of both concepts and entities, usually expressed by common and proper nouns, respectively.

To fill this gap, we present a methodology for automatically annotating a large corpus of sentences with CNER categories (Section 6.2.4), which we refer to as CNER$_{silver}$. Then, we describe the annotation procedure we adopted to produce CNER$_{gold}$, a manually-curated dataset for model evaluation (Section 6.2.5). Finally, in Section 6.2.6 we present a detailed analysis of our two resources compared to the existing ones.

6.2.4 Cner$_{silver}$

We propose an automatic approach to the creation of a large-scale training set for the CNER task. Our goal is, first, to disambiguate concepts and entities mentioned within text, and, second, to transform the resulting annotations into CNER categories. We chose the English Wikipedia as our corpus because it offers a large number of heterogeneous texts spanning all domains of knowledge. To ensure high quality, we restrict ourselves to the subset of articles that the Wikipedia community has deemed to be "good" or "featured"[1] and apply our annotation strategy to these articles.

Importantly, Wikipedia articles contain a few terms that are linked manually to other articles, but many other terms are left unlinked, leading to an issue of *sparsity*. Therefore, to ensure a high density of annotations, we perform three main steps:

- Because the Wikipedia guidelines specify that only the first occurrence of a term should be linked,[2] we propagate that link to all other occurrences of the term on the same page;

[1] Wikipedia Good and Featured Articles.
[2] Wikipedia Guidelines.

Category	Description	Examples
ANIMAL	Living beings (excluding humans) with the ability to move and perceive their surroundings.	dog, cat, brown bear, African wild dog, great white shark, *etc.*
ARTIFACT	Man-made objects, tools, and products.	vehicle, software, mouse, Windows XP, Fiat Panda, *etc.*
ASSET	Resources or possessions with economic or intrinsic value.	capital, stock, wealth, phone bill, Perkins Loan, *etc.*
BIOLOGY	Biological entities such as cells or organisms.	protein, cell, lipid, *E. coli*, herpes virus, *etc.*
CELESTIAL	Astronomical objects like planets, stars, and galaxies.	Sun, Neptune, Proxima Centauri, comet, *etc.*
CULTURE	Cultural aspects, traditions, or ideologies.	religion, feminism, socialism, Islam, Buddhism, *etc.*
DATETIME	Dates and temporal expressions.	18 March, Saturday, 1979, 15:30 AM, *etc.*
DISEASE	Medical conditions or disorders.	infection, allergy, acne, Alzheimer's disease, *etc.*
EVENT	Phenomena or activities at a specific time/-place.	crime, temperature change, Cannes Film Festival, *etc.*
FOOD	Edible items and beverages.	lasagna, carbonara, cheddar-beer fondue, *etc.*
LOC	Geographical locations and landmarks.	Rome, Mississippi River, Lake Paiku, *etc.*
ORG	Organizations and institutions.	Google, San Francisco Giants, Democratic Party, *etc.*
PER	Individuals or historical figures.	professor, musician, Ray Charles, Jessica Alba, *etc.*
PLANT	Types of trees, flowers, and other plants.	grass, peach tree, forsythia, *etc.*
STRUCT	Physical structures or constructions.	bridge, loft, St. Peter's Church, Golden Gate Bridge, *etc.*
SUPER	Mythological or religious entities.	Apollo, Persephone, Hercules, St. Peter, *etc.*

Table 6.1. CNER categories with concise descriptions and representative examples. For brevity, only a subset of categories and examples is shown, and we refer to the original paper for a more detailed table.

- As most of the mentions linked in Wikipedia articles refer to named entities, we ensure coverage and disambiguation of all common nouns with concepts in BabelNet (Navigli and Ponzetto, 2012; Navigli et al., 2021) through the application of Word Sense Disambiguation;
- All remaining entity mentions are tagged with standard NER.

Once each article has been fully annotated, we transform each tag, be it a Wikipedia hyperlink, a BabelNet concept, i.e. synset, or a NER category, into its CNER category. In what follows, we delve into the details of our *taxonomy-based tagging* strategy for annotating Wikipedia articles, and discuss the main heuristics for *solving sparsity*.

Taxonomy-based tagging To annotate each Wikipedia hyperlink w_j in an article W_i with a CNER category c, we first retrieve the corresponding BabelNet synset and map it to one of the 500 seed synsets introduced in Section 6.2.2. Specifically, given that each Wikipedia hyperlink corresponds to a synset node in the BabelNet taxonomy, to assign a CNER category to the hyperlink w_j, we start from the corresponding BabelNet node n_j and navigate upward along hypernymy edges within the hierarchy. When a seed synset with category c is reached, we assign c to the node n_j. In the case of multiple seed synsets reached at the same distance, we prioritize the most frequent category.

Solving sparsity As a result of the previous step, each hyperlink in a Wikipedia article is annotated with a CNER category. However, as previously mentioned, hyperlinks in Wikipedia are sparse (cf. line 1 in Table 6.2). To address this problem, we apply a surface matching heuristic where, for each link w_i with an associated category c and a surface text t_i, we propagate c to all text spans s in the same document such that $s = t_i \vee s \in syn(w_i)$, where $syn(w_i)$ is the set of synonyms of the BabelNet synset corresponding to w_i. However, while this methodology is remarkably effective for named entities, it falls short when it comes to densely annotating concepts. Indeed, after the surface matching heuristic, only 15% of common nouns are tagged, in contrast to 55.3% of proper nouns.

To fill this gap, we complement the above strategy with a state-of-the-art Word Sense Disambiguation model, ESCHER (Barba et al., 2021), to disambiguate each common noun in our dataset. As a result, we also obtain BabelNet synsets for the remaining unlinked common nouns, which we classify through the same taxonomy-based technique presented earlier. Finally, to annotate the remaining proper nouns with a CNER category, we use the Stanza NLP toolkit (Qi et al., 2020).

Sparsity heuristic	NOUN	PROPN
Wikilinks	7.1	33.2
+ surface matching heuristic	15.0	55.3
+ WSD	92.8	56.3
+ Stanza NER	97.3	98.4

Table 6.2. Impact of the various modules for solving sparsity in Wikipedia articles. The first row indicates the percentage of common and proper nouns hyperlinked in Wikipedia articles without applying any heuristic.

This toolkit produces named entity annotations using OntoNotes categories, which are then mapped to CNER categories by exploiting the one-to-one link between OntoNotes and CNER that was presented in Section 6.2.2. Remarkably, following the above-described process, we can annotate 98.4% of proper nouns and 97.3% of common nouns, hence obtaining extremely dense annotations. This is in contrast to prior work, as will be detailed later, in Section 6.2.6.

Table 6.2 reports an ablation study highlighting the distinct contributions of the previously described modules. Importantly, for each annotated span, we include explicit information on whether it is a concept (C) or a named entity (NE), which is essential for the training of the specialized NER and CR models, as will be detailed in Section 6.2.7. Since BabelNet provides this information for its synsets, we include it for each instance that is annotated via taxonomy-based tagging and WSD. The instances annotated using Stanza NER, instead, are all considered to be named entities except for dates and numbers. We then convert the dataset that has been produced employing the BIO tagging scheme.

6.2.5 Cner$_{gold}$

To enable a proper and rigorous evaluation of CNER models, we introduce an annotation procedure for constructing CNER$_{gold}$, a manually curated dataset. The annotation process started by formulating guidelines through a meticulous examination of examples within our CNER$_{silver}$ dataset. However, the task presented multiple challenges related to the annotation of multiword nominal expressions. First, we established that non-compositional expressions such as *white shark* and *bald eagle*, which represent specific concepts, have to be identified as whole spans,

Dataset	Type	# Sentences	# Tokens	# Spans	# Tagged Tokens	AVG # Spans	Density %
OntoNotes 5.0	Gold	76,714	1,388,973	104,151	190,310	1.36	13.69
UFET$_{crowd}$	Gold	5878	154,802	5994	17,627	1.01	11.38
UFET$_{silver}$	Silver	2,272,421	59,500,651	3,121,857	7,016,134	1.37	11.80
CNER$_{gold}$	Gold	2000	56,843	14,730	22,461	7.37	39.56
CNER$_{silver}$	Silver	317,590	8,725,846	2,282,800	3,403,952	7.18	39.00

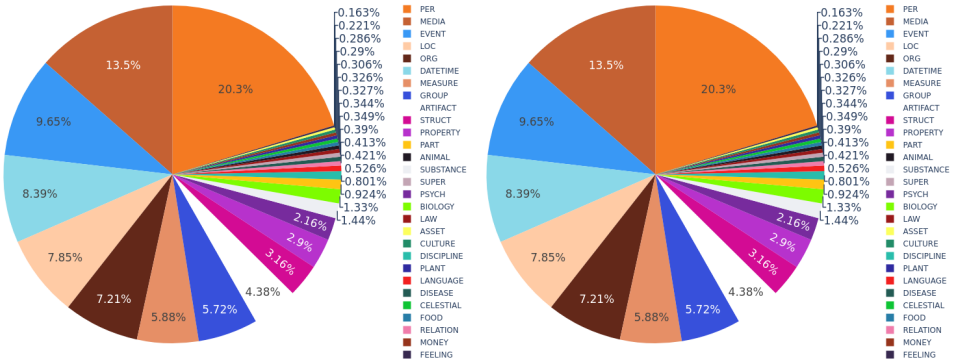
Table 6.3. Comparison between our newly-introduced CNER resource, OntoNotes, and UFET. AVG # Spans is the average number of tagged spans – either concepts or named entities – per sentence. Density % is the percentage of tagged tokens over the total number of tokens.

while in compositional expressions, such as *black laptop*, only the noun has to be tagged. Second, in the presence of nested named entities, e.g., *Mr. Smith goes to Washington*, we decided to tag the largest available span denoting an entity, rather than tagging, e.g., *Mr. Smith* and *Washington* separately. Third, we considered the cases in which the annotator is uncertain about whether a specific span of text is an entity or not (e.g. *One flew over the cuckoo's nest*), and what its potential meanings are (e.g. *mamihlapinatapai*). To help annotators disentangle these cases, we allowed them to use external world knowledge, i.e. giving access to web search, Wikipedia, etc., to properly identify and classify spans of text. We point out that this procedure proved to be effective for the annotation task: in a sample of 100 sentences, annotators, on average, utilized external resources for more than one span per sentence.

To start the annotation task, we randomly sampled 2,000 sentences from CNER$_{silver}$ and provided the annotators with a simple interface where each row displayed a specific token and its corresponding PoS tag. The annotators were asked to identify concepts and named entities by means of specific C vs NE tags and label them with the corresponding CNER categories by making their choice via a drop-down menu.

Because the dataset was annotated by a single expert linguist with a robust background in the annotation of lexical-semantic tasks, and an author of this work, we asked two other annotators to label 10% of the data. We then calculated the inter-annotator agreement on such instances and obtained a Fleiss' κ score of 89.8, indicating optimal agreement.

The overall annotation process took 2 months, with a total of 320 hours spent



(a) Pie chart showing the CNER$_{silver}$ category distribution.

(b) Pie chart showing the CNER$_{silver}$ category distribution.

Figure 6.3. CNER category distributions.

on the task. This process led to the creation of a corpus of 2,000 sentences and more than 56,000 annotated tokens, ∼22,000 of which were tagged with a CNER category.

6.2.6 Dataset Statistics

In Table 6.3, we provide the statistics of our CNER$_{gold}$ and CNER$_{silver}$ datasets compared to OntoNotes. We observe that our training set comprises a significantly larger number of annotated sentences, spans, and tokens. Furthermore, from the last two columns, we observe a considerably higher annotation density, highlighting that tagging both named entities and nominal concepts leads to a denser and richer annotation compared to focusing on entities only. In Table 6.3, we also present the statistics of UFET (Choi et al., 2018), but while their formulation encompasses the classification of both named entities and concepts, the annotation density is even lower than that of OntoNotes, as they disregard the identification part.

In addition, the pie charts in Figures 6.3a and 6.3b show the category balance in CNER$_{silver}$ and CNER$_{gold}$, respectively. Finally, to evaluate the quality of the automatic annotation procedure (Section 6.2.4), we compute the agreement over the set of 2,000 sentences reserved for CNER$_{gold}$. The Cohen's kappa coefficient over this set of samples is $\kappa = 71.4$, indicating a substantial agreement between the automatic and manual procedures. Figure 6.4, we provide the distribution of concepts and named entities for each category in our datasets.

6.2.7 Experimental Setup

We now describe the setup, including datasets (Section 6.2.7) and model architecture (Section 6.2.7). In our experiments, we aim to measure the performance of a competitive neural model on the CNER task and whether a CNER system performs on par with, or even better than, specialized NER and CR systems on the respective subtasks.

Datasets

In our experiments, we use several versions of our data:

- CNER$_{silver}$ and CNER$_{gold}$ are, respectively, the silver- and gold-standard datasets introduced in Section 6.2.7 covering both nominal concepts and named entities;
- NER$_{silver}$ and NER$_{gold}$ are the same datasets as above, in which only entity-related annotations are retained (i.e., concepts are mapped to the O class);
- CR$_{silver}$ and CR$_{gold}$ are the same datasets as above, in which only concept-related annotations are retained (i.e., named entities are mapped to the O class);

Based on these datasets, we train and evaluate the same model architecture under the following four different settings: i) we train on CNER$_{silver}$ and test on CNER$_{gold}$, ii) we train on CNER$_{silver}$ and test on NER$_{gold}$ and CR$_{gold}$, iii) we train on NER$_{silver}$ and test on NER$_{gold}$, and iv) we train on CR$_{silver}$ and test on CR$_{gold}$.[3] Different from setting (i) in which a distinction between concepts and named entities is not required, in setting (ii), we train a CNER model on CNER$_{silver}$ which, for each predicted span, provides the type (C or NE) in addition to the CNER category (e.g., for the token *doctor* the model outputs *B-C-Person*). This allows us to retain only entity- or concept-related annotations when evaluating on NER$_{gold}$ and CR$_{gold}$, respectively, enabling a fair evaluation and ensuring comparability between the CNER model

[3]We split the gold data into half for validation, half for testing (we refer to the latter as CNER$^{test}_{gold}$, NER$^{test}_{gold}$ and CR$^{test}_{gold}$).

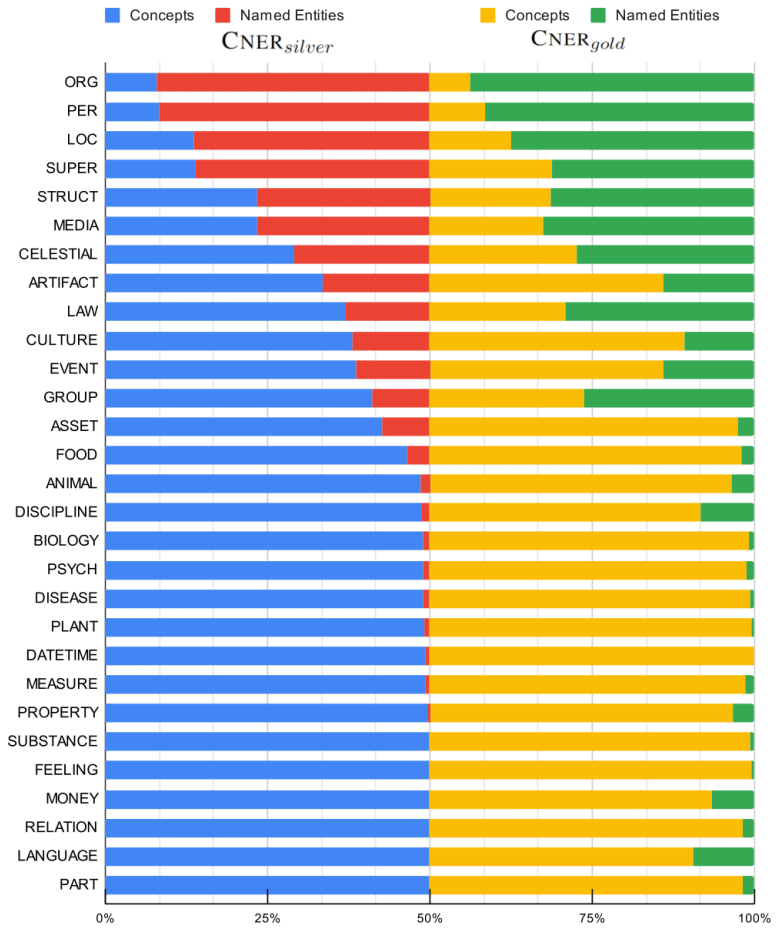


Figure 6.4. Bar chart with the distribution of named entities and concepts for every CNER category in the CNER$_{gold}$ and CNER$_{silver}$ resources.

and the specialized NER and CR systems.[4] In fact, an unfiltered evaluation of CNER predictions would lead to uninformative results, since a CNER model would naturally provide dense annotations regarding all the nominal expressions in a sentence, independently of their type (C or NE).

As evaluation metrics, we adopt the macro F1 score at the token level and the span-based F1 score. Additionally, we also report token-level micro F1 scores, because – differently from NER – the O category is much less frequent, given the high density of CNER annotations (cf. last column of Table 6.3).

[4]We highlight that NER systems implicitly learn to identify NEs, in the same manner as CR does with concepts.

Architecture

For our experiments, we opted for an architecture that strikes a balance between accuracy and simplicity, offering the robustness required for our task while being efficient and easy to fine-tune. Specifically, we employ a pretrained transformer encoder, namely DeBERTa-v3 base (He et al., 2023), and train a classification head on top of it to predict the category for each token in a sentence. The classification head consists of two linear layers and a normalization layer, with a dropout of 0.1 and the GeLU activation function. Since this architecture has been proven to achieve competitive performance compared to the current state of the art in standard NER benchmarks (He et al., 2023), we believe that it constitutes a strong baseline for the CNER task.

Our systems are developed using the PyTorch Lightning framework[5] and the HuggingFace models library.[6] We train them on a single RTX 4090 Ti, with a patience parameter of 20 and a validation step every 30% of the total number of steps per epoch, resulting in 4 epochs and approximately 2 hours of training time for each model. We use the RAdam optimizer with a learning rate of 5×10^{-6}, and a linear scheduler with a warm-up of 10% of the total steps. We adopt the same architectural setup and hyperparameters to train the CNER, NER, and CR models. All models are trained to minimize the cross-entropy loss function. We select the best model based on its macro F1 score on the validation set.

6.2.8 Results

CNER performance

The proposed architecture achieves 87.20 micro F1, 59.38 macro F1, and 66.72 span F1 score points, when asked to identify and classify both named entities and concepts with our 29-category tagset (cf. setting (i) in Section 6.2.7). In Figure 6.5, we provide the individual span F1 scores for each category in our benchmark. In general, by cross-referencing the category scores with the distribution in Figure 6.3a, we observe that performance has a moderate correlation with the number of training

[5] https://www.pytorchlightning.ai/
[6] https://huggingface.co/docs/transformers/

Test / Train	NER^{test}_{gold}			CR^{test}_{gold}		
	Micro	Macro	Span	Micro	Macro	Span
(ii) CNER_{silver}	**94.07**	**39.65**	**69.15**	**91.41**	**52.47**	**63.77**
(iii) NER_{silver}	93.59	34.28	66.73	—	—	—
(iv) CR_{silver}	—	—	—	89.71	44.48	59.66

Table 6.4. Micro, Macro and Span F1 scores (%). (ii), (iii), and (iv) refer to the settings described in Section 6.2.7.

instances (Pearson correlation coefficient $\rho = 0.4$).

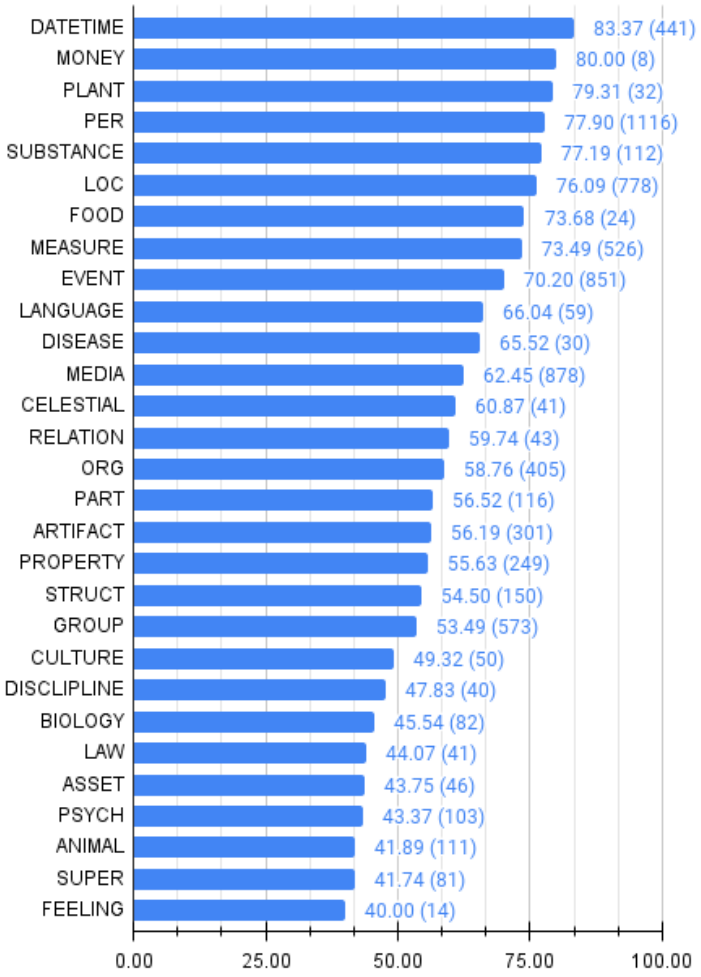


Figure 6.5. Bar chart with span F1 score (%) for each CNER category sorted in descending order. The support for each category is in parentheses.

From a closer perspective, our system is affected by two main factors. First, when evaluated only on the identification of spans, the system achieves 86.83 span F1 score points, setting an upper bound to the overall system performance. Indeed, the model struggles to identify spans like *Triple J hottest 100 of all time in Australia* and *A full day's Work* in their entirety. Second, the confusion matrix Figure 6.6 shows that the model sometimes mistakes categories that have an intrinsic level of ambiguity (e.g. ANIMAL or PLANT with FOOD). In particular, terms that can be associated with multiple categories (e.g. *chicken* and *eggplant*) are often difficult to classify based on the limited sentence context. Additionally, specific terms, often contained in Wikipedia articles, like *Synapsids Ophiacodon* or *metal umlaut*, require technical expertise to be properly classified. Similarly, the categories PROPERTY, PSYCH, and FEELING are frequently confused due to their intertwined semantics, while instances belonging to the LAW category, such as *peace treaty*, can occasionally overlap with instances from the MEDIA category, which encompasses general written documents conveying information.

CNER vs NER vs CR

We summarize the results of our experiments in Table 6.4. Our findings clearly illustrate the significant synergistic benefits of training a single model to simultaneously identify and classify concepts and named entities, compared to restricting the same model to either concepts or named entities. The CNER model exhibits a notable increase of ∼0.5 micro, ∼5.4 macro, and ∼2.4 span F1 points over the NER model. A similar improvement is achieved over the CR model with a ∼1.7 micro, ∼8.0 macro, and ∼4.1 span F1 point increase. These findings underline the advantages of our proposed CNER approach in improving the accuracy and effectiveness of both concept and named entity recognition tasks. In the next Section, we provide a qualitative analysis of our results.

6.2.9 Qualitative Analysis

To better understand the behavior of CNER, NER, and CR models, we conduct a qualitative error analysis on $\text{CNER}^{test}_{gold}$ and report some representative examples in

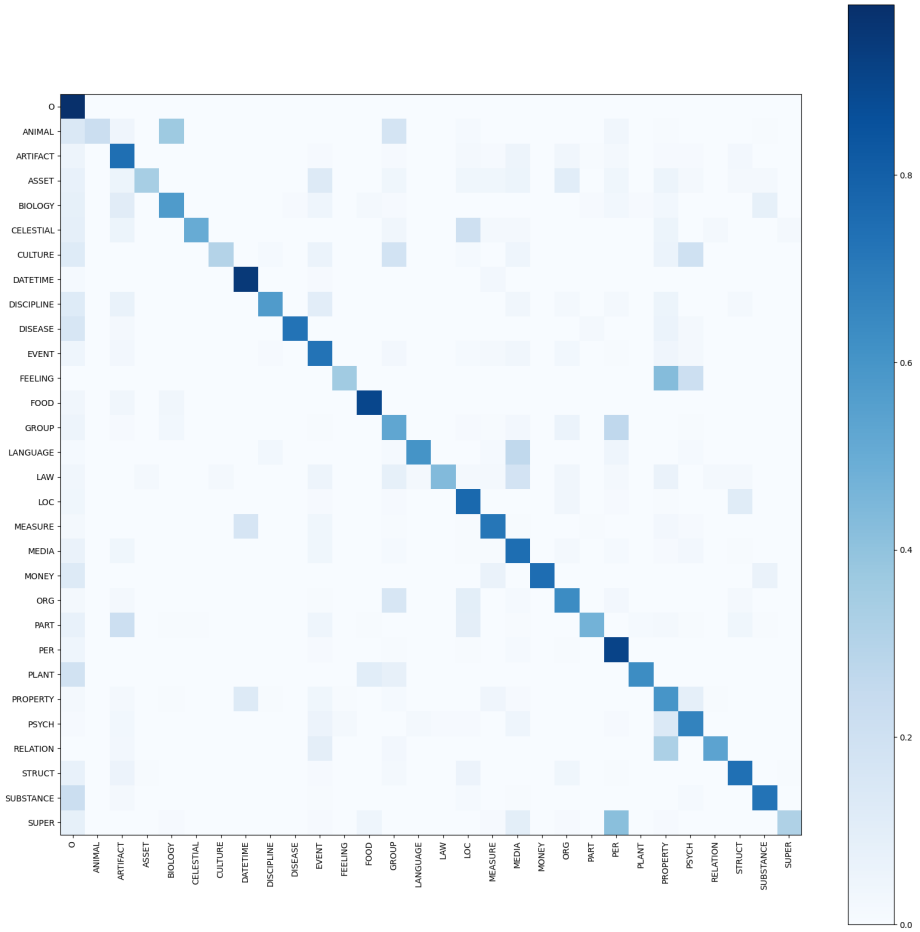


Figure 6.6. Confusion matrix on the CNER task.

Table 6.5.

In the first example, we can immediately appreciate the higher annotation density of both concepts and named entities produced by the CNER model compared to the specialized NER and CR alternatives. We also report, in the second example, an instance in which the CNER model correctly predicts the label PROPERTY for the concept *role* while CR misclassifies it as an EVENT. Furthermore, in the same sentence, we have an example of an instance for which there is one correct label, but, based on the given context, other tags could be associated with it. Specifically, both the CNER and CR models misclassify *tobacco* by tagging it with FOOD and SUBSTANCE, respectively, which are less appropriate, but both plausible annotations within the given context. In the third example, instead, we present a sentence in which our system shows better generalization capabilities: while the named entity

	Type	Prediction
Example 1	Gold	The [**first**]$_{MEASURE}$ [**town**]$_{LOC}$ to fall into the [**Lombards**]$_{GROUP}$' [**hands**]$_{PART}$ was [**Forum Iulii**]$_{LOC}$ ([**Cividale del Friuli**]$_{LOC}$), the [**seat**]$_{LOC}$ of the local [**magister militum**]$_{PER}$.
	NER	The first town to fall into the [**Lombards**]$_{GROUP}$' hands was [**Forum Iulii**]$_{STRUCT}$ ([**Cividale del Friuli**]$_{LOC}$), the seat of the local magister militum.
	CR	The [**first**]$_{MEASURE}$ [**town**]$_{LOC}$ to fall into the Lombards' [**hands**]$_{PART}$ was Forum Iulii (Cividale del Friuli), the [**seat**]$_{LOC}$ of the local [**magister**]$_{PER}$ militum
	CNER	The [**first**]$_{MEASURE}$ [**town**]$_{LOC}$ to fall into the [**Lombards**]$_{GROUP}$' [**hands**]$_{PART}$ was [**Forum Iulii**]$_{STRUCT}$ ([**Cividale del Friuli**]$_{LOC}$), the [**seat**]$_{LOC}$ of the local [**magister**]$_{PER}$ militum
Example 2	Gold	[**Tom McCarthy**]$_{PER}$ highlighted the prominent [**role**]$_{PROPERTY}$ of [**tobacco**]$_{PLANT}$ in the [**story**]$_{MEDIA}$, drawing on the [**ideas**]$_{PSYCH}$ of [**philosopher Jacques Derrida**]$_{PER}$ to suggest the potential [**symbolism**]$_{MEDIA}$ of this.
	NER	[**Tom McCarthy**]$_{PER}$ highlighted the prominent role of tobacco in the story, drawing on the ideas of [**philosopher Jacques Derrida**]$_{PER}$ to suggest the potential symbolism of this.
	CR	Tom McCarthy highlighted the prominent [**role**]$_{EVENT}$ of [**tobacco**]$_{SUBSTANCE}$ in the [**story**]$_{MEDIA}$, drawing on the [**ideas**]$_{PSYCH}$ of philosopher Jacques Derrida to suggest the potential [**symbolism**]$_{PSYCH}$ of this.
	CNER	[**Tom McCarthy**]$_{PER}$ highlighted the prominent [**role**]$_{PROPERTY}$ of [**tobacco**]$_{FOOD}$ in the [**story**]$_{MEDIA}$, drawing on the [**ideas**]$_{PSYCH}$ of [**philosopher Jacques Derrida**]$_{PER}$ to suggest the potential [**symbolism**]$_{PSYCH}$ of this.
Example 3	Gold	[**John**]$_{SUPER}$ has new [**weapons**]$_{ARTIFACT}$, including [**holy water**]$_{SUBSTANCE}$, [**bait**]$_{BIOLOGY}$ for the [**undead**]$_{SUPER}$, and a [**blunderbuss**]$_{ARTIFACT}$ that uses [**zombie**]$_{SUPER}$ [**parts**]$_{ARTIFACT}$ as [**ammunition**]$_{ARTIFACT}$.
	NER	[**John**]$_{PER}$ has new weapons, including holy water, bait for the undead, and a blunderbuss that uses zombie parts as ammunition.
	CR	John has new [**weapons**]$_{ARTIFACT}$, including [**holy water**]$_{SUBSTANCE}$, [**bait**]$_{SUBSTANCE}$ for the [**undead**]$_{BIOLOGY}$, and a blunderbuss that uses [**zombie**]$_{BIOLOGY}$ [**parts**]$_{ARTIFACT}$ as [**ammunition**]$_{ARTIFACT}$.
	CNER	[**John**]$_{PER}$ has new [**weapons**]$_{ARTIFACT}$, including [**holy water**]$_{SUBSTANCE}$, [**bait**]$_{SUBSTANCE}$ for the undead, and a blunderbuss that uses [**zombie**]$_{SUPER}$ [**parts**]$_{ARTIFACT}$ as [**ammunition**]$_{ARTIFACT}$.

Table 6.5. Examples of annotations from the inference of our CNER, NER, and CR models over the test split of our CNER$_{gold}$ dataset. Correct predictions are highlighted in green, while wrong predictions are highlighted in red.

John is misclassified as PER by both the NER and CNER models, the CNER system correctly assigns the label SUPER to the concept *zombie*, which is misclassified by the CR model with the tag BIOLOGY.

Additionally, in Figure 6.7, we illustrate the category-wise impact of the CNER system when compared to NER and CR systems, highlighting its positive and negative contributions. Notably, the CNER system consistently exhibits a positive influence over both named entities and concepts. Specifically, categories such as DISCIPLINE, FOOD, and CULTURE witness particularly positive contributions, with an increase of up to 40 span-based F1 score points. In some categories, the CNER system provides positive contributions only for named entities, as observed in GROUP and MEDIA, or exclusively for concepts, as evidenced in SUPER and LAW.

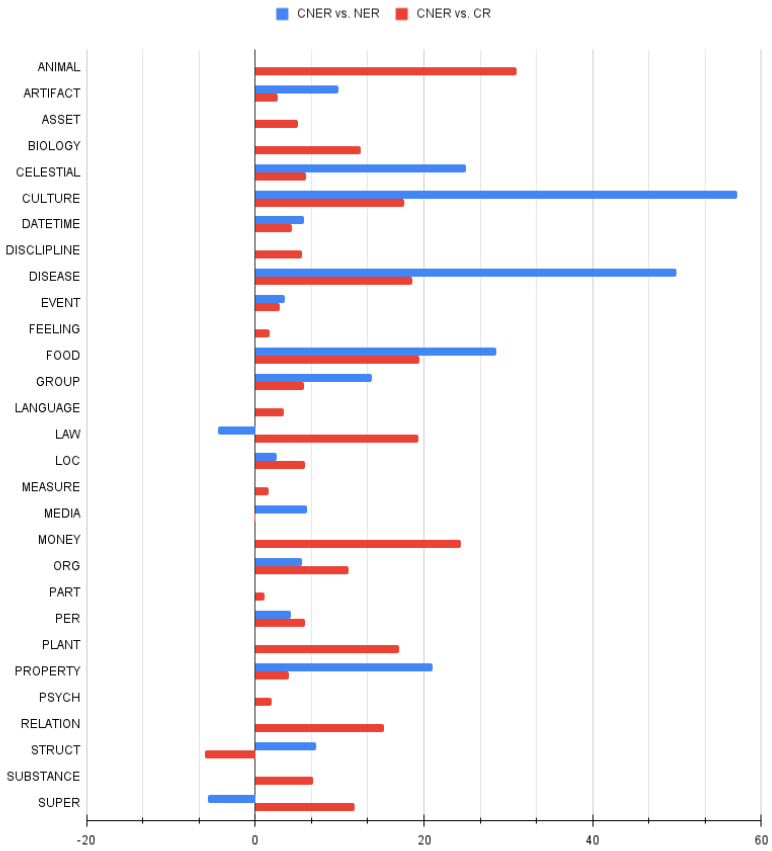


Figure 6.7. Bar chart with Span F1 score of the different categories.

6.3 Semantically Typed Coreference Resolution

In the previous Section, we introduced the CNER framework, which provides a unified inventory of semantic types covering both *named entities* (e.g., proper nouns such as *Barack Obama*, *London*) and *nominal concepts* (e.g., common nouns such as *the president*, *the city*). We have shown that a simple encoder-only Transformer architecture, a CNER model, can identify spans of text and classify them into those categories with high performance. As we anticipated in Section 6.1, this coarse-grained semantic annotation can be adopted for tagging coreference outputs, since they often include a mix of proper-noun and common-noun mentions, along with pronominal entities.

Background and Motivation. Some previous works have already explored the connection between semantic typing and Coreference Resolution. Early feature-based systems exploited shallow semantic cues, such as semantic class compatibility or lexical relations, to improve mention linking and constrain implausible antecedents (Ng, 2007). Subsequent structured models explicitly coupled *entity typing*, *entity linking*, and *coreference* in joint formulations, demonstrating that shared semantic representations yield more coherent and interpretable clusters (Durrett and Klein, 2014; Chen et al., 2017; Agarwal et al., 2022). With the rise of neural architectures, type information began to be incorporated directly into mention or span representations, either leveraging predicted NER embeddings or dedicated classification heads (Clark and Manning, 2015; Khosla and Rose, 2020). More recently, Mtumbuka and Schockaert (2024) has shown that coreference relations themselves can serve as distant supervision for fine-grained entity typing, underscoring the bidirectional synergy between the two tasks.

Despite these promising developments, most prior work remains constrained by the limited capabilities of conventional NER taxonomies. Typical systems rely on a handful of broad, well-established NER categories (PERSON, ORGANIZATION, LOCATION), which, while effective for named entities, provide insufficient coverage and semantic granularity for the diverse mention types encountered in realistic texts. Furthermore, those methods are usually based on standard NER practices, which only encompass the categorization of named entities and ignore common nouns or abstract mentions, leading to sparse or shallow semantic typing.

CNER-based Semantically Typed Coreference. By leveraging CNER, we now propose an advancement to this line of research under a richer semantic condition. Unlike standard NER frameworks, CNER offers a more comprehensive and finer-grained ontology covering both named entities and common nouns. This expanded coverage allows for a more principled and interpretable form of semantically typed coreference, in which clusters can be automatically assigned a meaningful semantic label that reflects their underlying concept.

Our approach adopts a neural CNER model to tag all nominal mentions in

the corpus and then propagates these mention-level labels to the coreference clusters using an overlap-based heuristic. The resulting clusters are annotated with consistent semantic types, facilitating a richer analysis of model behavior across domains and error categories. The only notable exception occurs for clusters consisting solely of pronominal mentions, which cannot be directly tagged by CNER; however, such cases are infrequent and have a negligible effect on overall coverage, as we show in Section 6.5.

6.3.1 Propagation heuristic

CNER categories are propagated to coreference clusters via a two-step procedure, composed of *Mention Assignment* and *Category Propagation*.

1. **Mention Assignment.** In this step, we tag each mention predicted by the coreference model with a CNER label based on token overlap. We only assign a label to named entities and common nouns, excluding pronouns, which are outside the CNER taxonomy. Each mention inherits the label of the CNER span with which it has the highest overlap.

2. **Category Propagation.** In this step, we first assign a semantic category to the coreference cluster in which at least one mention was tagged with a CNER class. Then, for each of these clusters, we propagate this label to all the coreferent mentions. To assign a cluster-level semantic type, we examine all mentions in a cluster and select the category that appears most frequently. This majority-vote approach captures the representative semantic type of the entity and is subsequently propagated to any previously unlabeled mentions, such as pronouns, thereby completing the semantic labeling for all mentions in the cluster.

Formal Definition

Let D be a document with a set of mentions $M = \{m_1, m_2, \ldots, m_n\}$ predicted by a coreference model, and a set of CNER-annotated spans $C = \{c_1, c_2, \ldots, c_k\}$ with semantic labels $\mathcal{L}(c_i) \in \mathcal{T}$, where $\mathcal{T}$ is the inventory of CNER categories defined in

Section 6.2.2.

Each textual element, either coreference mentions m_j or CNER annotations c_i, is represented by its token span, denoted span(x). We define the overlap function $\Omega(m_i, c_j)$ as the token-level Jaccard similarity between the mention m_i and a CNER span c_j:

$$\Omega(m_i, c_j) = \frac{|\text{span}(m_i) \cap \text{span}(c_j)|}{|\text{span}(m_i) \cup \text{span}(c_j)|},$$

This measures how closely a mention aligns with a semantically labeled span, normalized by their combined token coverage.

Mention Assignment. In the first step, each mention m_i is assigned the label l_i of the overlapping CNER span with the highest Ω score, provided that $\Omega(m_i, c_j) \geq \tau$, where τ is an empirical threshold (set to 0.5 in our experiments). Mentions without sufficient overlap are labeled as $l_i =$UNLABELED.

Category Propagation. In the second step, we assign a semantic label to each coreference cluster based on the mentions it contains. Let $\mathcal{G} = \{G_1, G_2, \ldots, G_r\}$ denote the set of predicted clusters. For each cluster G_v that contains at least one labeled mention, we determine its cluster-level label via majority voting over the mention labels. Formally, for each label $t \in \mathcal{T}$, we count how many mentions m_z in G_v are tagged with $l_z = t$, and select the most frequent label:

$$\mathcal{S}(G_v) = \arg\max_{t \in \mathcal{T}} |\{m_z \in G_v \mid l_z = t\}|.$$

If two or more labels occur with equal frequency, we break ties by selecting the label whose mentions have, on average, the highest overlap Ω with their corresponding CNER spans. Once the dominant label $\mathcal{S}(G_v)$ is determined, it is propagated to all mentions in $m_z \in G_v$ that were previously unlabeled.

6.4 Experimental Setup

In our experiments, we aim to showcase how adding coarse-grained semantic typing to coreference outputs improves the interpretability of model errors, especially

for assessing the *cross-domain robustness* of coreference models, under the lens of semantic categories. Specifically, we investigate the capabilities of models trained on one dataset to generalize to different domains and whether semantic typing exposes domain-specific weaknesses that are otherwise obscured by aggregate metrics. We first report the results of our propagation heuristic across datasets, followed by analyses of model performance and error patterns after applying semantic typing. We now introduce datasets and models used to perform this analysis.

6.4.1 Datasets

We conduct experiments on three Coreference Resolution datasets:

- **OntoNotes** (Pradhan et al., 2012). Includes documents spanning seven heterogeneous genres such as newswire, broadcast news, magazines, weblogs, conversational telephone speech, and religious texts. Its relatively broad topical coverage makes it the de facto standard for evaluating general-domain CR models. OntoNotes' mentions include a variety of noun types (proper nouns, common-noun mentions, pronominal mentions), and its multi-genre nature implies a broader and more balanced distribution of entity types than a single-genre literary corpus.

- **LitBank** (Bamman et al., 2020) is annotated according to six ACE-style entity categories: PER (persons), FAC (facilities), LOC (locations), GPE (geopolitical entities), ORG (organizations), and VEH (vehicles). From their original report, about 83.1% of the mentions in LitBank are persons, reflecting the narrative emphasis on characters in literary texts. Other types appear with much lower frequency: facilities account for about 8.0%, locations 4.4%, geopolitical entities 3.3%, vehicles 0.7%, and organizations 0.5% (Bamman et al., 2020).

- **PreCo** (Chen et al., 2018) domain (reading comprehension texts) tends to emphasize common and named entities in everyday contexts, yielding a more balanced distribution across categories than literary corpora.

6.4.2 Comparison Systems

We compare the following three variants of Maverick, each fine-tuned on a different dataset, presented in Section 3.4:

- **maverick-mes-ontonotes**[7]: a Maverick model fine-tuned on the OntoNotes dataset, based on DeBERTa-large. It currently has the best performance on the CoNLL-2012 test set.
- **maverick-mes-litbank**[8]: a Maverick model fine-tuned on the LitBank dataset. It also uses the same DeBERTa-large architecture and obtains 78.0 CoNLL-F1 on the LitBank.
- **maverick-mes-preco**[9]: a Maverick model fine-tuned on the PreCo dataset. This version uses the same DeBERTa-large and achieves 87.4 CoNLL-F1 on PreCo.

These three models allow us to examine how training on different domains influences performance, cross-domain generalization, and the behavior of semantically typed coreference. Because they share the same base architecture but differ in training data and annotation conventions, the comparison isolates the effect of domain and annotation scheme shifts.

6.4.3 Metrics

To evaluate semantically-typed coreference resolution performance, we propose to use two metrics that can be adapted well to measure the identification of entity mentions and coreference links, both globally and within specific semantic categories: Mention F1 and Link F1.

Mention F1. Mention F1 quantifies the quality of mention extraction independently of clustering. It is defined as a traditional F1 score, i.e., a harmonic mean of mention precision and recall, where precision measures the proportion of predicted

[7] `https://huggingface.co/sapienzanlp/maverick-mes-ontonotes`
[8] `https://huggingface.co/sapienzanlp/maverick-mes-litbank`
[9] `https://huggingface.co/sapienzanlp/maverick-mes-preco`

mentions that exactly match gold-standard mentions, and recall measures the proportion of gold mentions that are correctly predicted. In addition to overall Mention F1, we report scores stratified by semantic category (e.g., PER, ORG, STRUCT), which allows us to analyze systematic biases or disparities in mention detection across different types.

Link F1. Link F1 evaluates the quality of predicted coreference links between mentions. Each coreference cluster—gold or predicted—is represented as the set of all mention pairs that co-occur within the same cluster. By measuring predicted coreference capabilities in terms of F1 score over these two sets, Link F1 measures how accurately the model reconstructs reference relations among mentions. Unlike Mention F1, which isolates span detection, Link F1 captures the structural quality of clustering. In our experiments, we compute Link F1 using gold mentions to control for mention detection effects and to focus exclusively on the model's linking performance.

Interpretation. Together, Mention F1 and Link F1 provide a coherent and interpretable decomposition of coreference performance. Mention F1 captures *what* the model recognizes as entity mentions, while Link F1 captures *how* these mentions are connected into coherent referential structures. When computed per semantic class, these metrics enable a fine-grained diagnostic view of model behavior across different entity types, revealing both linguistic coverage and potential biases in the learned representations.

6.5 Results

In this section, we first analyze the coverage and category distribution of our CNER propagation heuristic, then present model performance with per-class analyses based on Mention F1 and Link F1.

6.5.1 Semantically-Typed Coreference Propagation Heuristic

Coverage Table 6.6 presents the coverage of our CNER propagation heuristic described in Section 6.3. We report the proportion of gold mentions in each dataset for which we can assign a CNER label, distinguishing between the two stages: Mention Assignment, which denotes the share of mentions directly labeled by our initial span-overlap heuristic, and Category Propagation, which represents the overall coverage after label propagation within clusters using the majority-vote strategy.

Dataset	(1) Mention Assignment	(2) Category Propagation
OntoNotes	53.3%	89.1%
LitBank	37.5%	88.0%
PreCo	71.4%	87.0%

Table 6.6. Coverage of the CNER propagation heuristic across datasets. *Mention Assignment* refers to the proportion of mentions directly labeled by the span-overlap heuristic, while *Category Propagation* shows the proportion of mentions that receive a label after cluster-level propagation using majority voting.

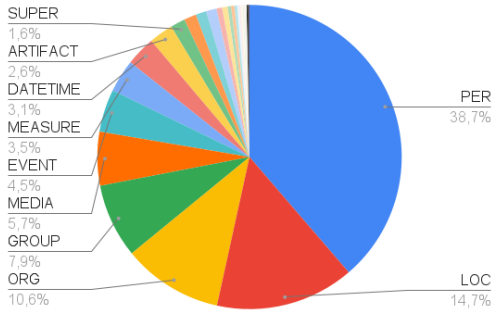
After the Mention Assignment step, coverage varies substantially across datasets. Datasets with fewer pronoun mentions, such as PreCo, tend to have higher initial coverage (71.4%), whereas datasets like LitBank, which contain many pronominal references typical of literary texts, show lower coverage (37.5%).

The subsequent Category Propagation step significantly increases coverage: the vast majority of mentions are successfully labeled, demonstrating that our heuristic effectively propagates semantic types even to initially unlabeled mentions, including pronouns.

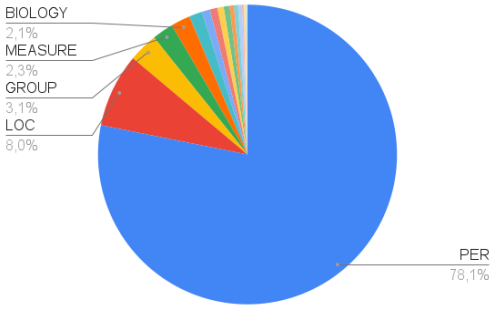
Nonetheless, roughly 10% of mentions remain unlabeled in each dataset. A careful qualitative inspection shows that these correspond to clusters composed exclusively of pronouns, which cannot be directly tagged by CNER. Handling these could be addressed in future work by training a simple classification method that is weakly supervised on our assigned labels, but here we restrict evaluation to the labeled portion.

Categories distribution To better understand the semantic composition of each corpus, we compute the distribution of CNER semantic categories across OntoNotes,

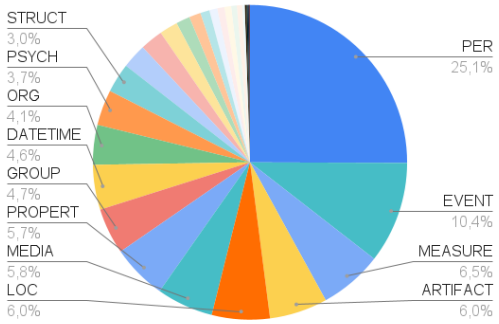
LitBank, and PreCo, as shown in Figure 6.8. This analysis highlights how domain and genre shape the prevalence of entity types, providing context for interpreting cross-domain coreference results that will follow in the next sections.



(a) OntoNotes gold mentions' category distribution after propagating CNER labels.



(b) LitBank gold mentions' category distribution after propagating CNER labels.



(c) PreCo gold mentions' category distribution after propagating CNER labels.

Figure 6.8. Category distributions on our datasets

First, we note that the category distribution produced by our propagation heuristic in LitBank closely follows the distribution reported by the dataset authors in Section 6.4.1, providing an extrinsic validation of our method. However, this also highlights the skewed nature of LitBank annotations: PER mentions dominate due to character-centered narratives, while categories such as LOC and ORG appear only marginally. Additionally, in LitBank, several categories of mentions are completely absent, including MONEY, PLANT, ASSET, FOOD, DISCIPLINE, CELESTIAL, LANGUAGE, DISEASE, RELATION, LAW, and CULTURE. As noted by the dataset authors, this is because LitBank annotation focuses only on a subset of semantic classes, leaving certain types unannotated. This limitation motivates our subsequent evaluation of cross-domain capabilities: a model trained on

LitBank may struggle to resolve coreference for mentions of these missing semantic categories.

In contrast, OntoNotes, known for its multi-genre coverage, exhibits a more balanced distribution across semantic categories, with substantial proportions of PER, ORG, LOC, and EVENT mentions. PreCo also demonstrates a balanced representation of classes, with only 25% of its mentions referring to people and a more diverse distribution of the other classes, particularly focusing on events and artifacts.

These patterns illustrate a core challenge in cross-domain evaluation: models trained on highly person-centric corpora like LitBank may overfit to human-referential semantics, while models trained on OntoNotes or PreCo generalize more evenly across diverse entity types, including organizations, events, and abstract categories.

6.5.2 Model Results

We evaluate the three variants of Maverick, each fine-tuned on one of OntoNotes, LitBank, or PreCo, and assess their performance both in-domain and cross-domain, i.e., either tested on the same dataset they were trained on or on a different dataset. Table 6.7 reports the micro-averaged Mention F1 scores across all configurations.

Model	OntoNotes	LitBank	PreCo
maverick-mes-ontonotes	**89.8**	74.3	49.4
maverick-mes-litbank	67.1	**91.6**	49.8
maverick-mes-preco	70.3	64.2	**93.2**

Table 6.7. Micro-averaged Mention F1 scores for each comparison system evaluated on all of our three datasets.

As expected, all the models perform very well in-domain, achieving high Mention F1 scores. However, when applied to out-of-domain corpora, performance drops substantially, confirming the difficulty of transferring mention extraction capabilities across datasets that differ in genre and annotation conventions.

To better interpret these results from a semantic perspective, Figure 6.9 reports the per-class Mention F1 for each model.

For clarity, we show both in-domain results (on the test set of the dataset used for training) and out-of-domain results, computed as the average across the

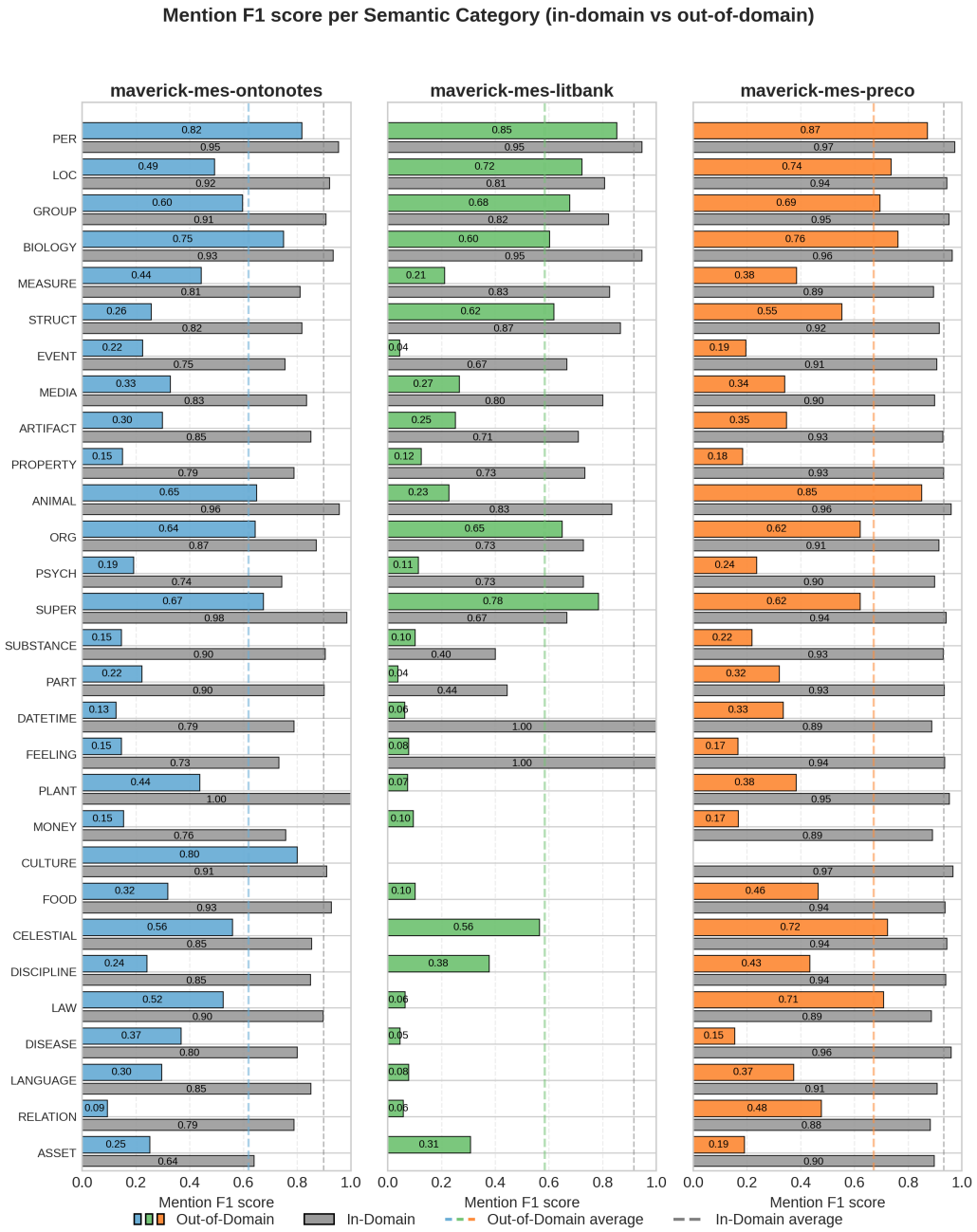


Figure 6.9. Per class Mention F1 for each model. We report in-domain results in grey, and use different colours to report out-of-domain results, computed as the average across the scores obtained in the two datasets on which the model was not trained on. We also plot the averaged value in the domain and out-of-domain results as dashed vertical lines. Classes are sorted with decreasing order with respect to category distribution in LitBank to further highlight cross-domain transfer performances.

two datasets on which the model was not trained.

Several patterns clearly emerge. First, all models maintain strong in-domain performance across most classes, consistent with the aggregate micro-averaged Mention F1 scores reported in Table 6.7. We also note that, for LitBank, some categories are missing in the in-domain evaluation as they were not identified in that corpus.

Second, out-of-domain performance reveals pronounced discrepancies across models and semantic classes. The LitBank-trained model, in particular, struggles to generalize to other datasets, showing a steep performance decline when applied to non-literary texts. Nonetheless, well-represented classes such as PER, LOC, and GROUP maintain relatively high performance across domains and models. This confirms that training data distribution heavily shapes model behavior, and annotation guidelines do not vary a lot when dealing with such easily detectable classes. This is not true for other classes, such as LAW, DISEASE, MONEY, PLANT, and FEELING, as they are less present in the datasets and their span annotation might be more dependent on specific annotation guidelines.

In fact, we hypothesize that this lack of cross-domain transfer arises from two main factors:

1. **Annotation guidelines:** Models tend to follow corpus-specific annotation conventions and labeling patterns, which can lead to systematic biases when evaluated under different annotation schemes.

2. **Coverage of semantic categories:** Underrepresented or rare entity types in the training corpus are harder for models to recognize and extract, resulting in inconsistent performance across domains.

To disentangle these two effects, we conduct a secondary analysis focusing exclusively on the coreference linking step. In this setting, we use gold mentions as input and evaluate only the models' ability to correctly link mentions within clusters. We compute Link F1 scores (excluding singletons) to measure linking accuracy independently of mention detection. Results are reported in Table 6.8.

From Table 6.8, we observe that, when isolating linking capabilities, the model

Model	OntoNotes	LitBank	PreCo
maverick-mes-ontonotes	**91.8**	**89.1**	76.5
maverick-mes-litbank	79.9	80.2	81.8
maverick-mes-preco	81.6	77.3	**94.3**

Table 6.8. Micro-averaged Link F1 scores starting from gold mentions.

fine-tuned on OntoNotes achieves the highest linking scores, both in-domain and when evaluated on LitBank, suggesting a strong degree of cross-domain robustness. This behavior likely stems from OntoNotes' wide genre diversity and balanced entity distribution, which allow the model to develop a more generalizable linking strategy. By contrast, the LitBank-trained model, despite its excellent mention extraction performance on its own corpus (cf. Table 6.7), transfers poorly to other domains. This finding is in line with previous works that show presence the underscores how narrower narrative-style training data and more restrictive annotation guidelines can lead to overfitting to specific textual characteristics and mention patterns. Finally, the PreCo-trained model performs extremely well in-domain but only moderately across other corpora, reflecting its focus on simple texts that substantially differ from the more complex genres of OntoNotes of the literary style in LitBank.
To further probe these dynamics, we analyze per-class linking performance across both in-domain and out-of-domain conditions (Figure 6.10).

The figure confirms that classes with limited representation in the training data, located toward the bottom of the chart, systematically achieve lower Link F1 scores. This effect is particularly visible for the LitBank-trained model, which underperforms for most non-PER categories, while models trained on OntoNotes and PreCo exhibit a more uniform distribution of linking quality.

Overall, our findings suggest that domain shift in coreference resolution cannot be fully understood or addressed without considering the semantic composition of the training data. A model trained on data with a narrow entity type distribution may appear competitive on standard metrics but will likely fail to capture the full diversity of referential phenomena present across domains. Future work should therefore prioritize corpus designs that balance annotation consistency with broader coverage of semantic categories.

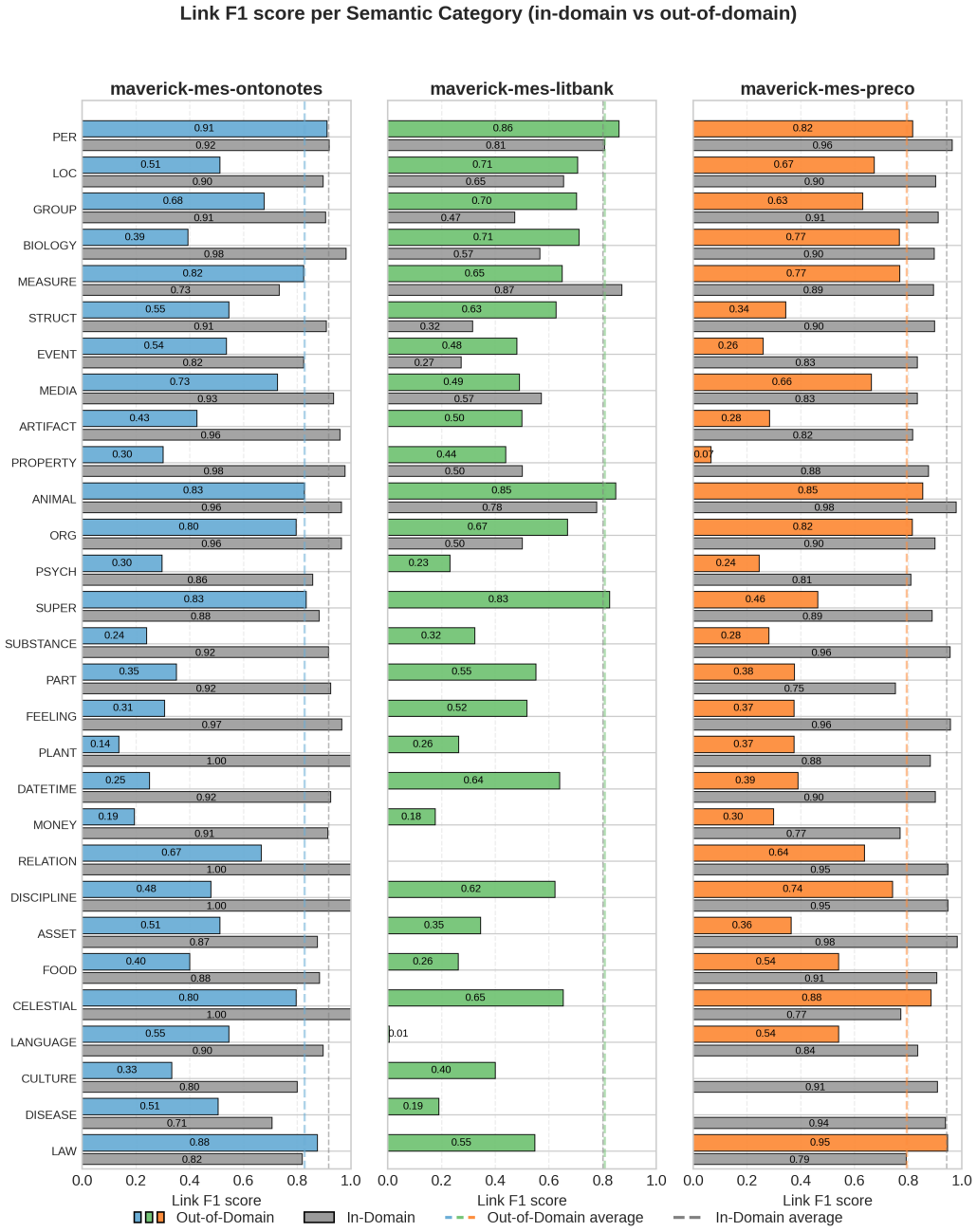


Figure 6.10. Per class Link F1 starting from gold mentions for each model. We report in-domain results in grey, and use different colours to report out-of-domain results, computed as the average across the scores obtained in the two datasets on which the model was not trained on. We also plot the averaged value in the domain and out-of-domain results as dashed vertical lines. Classes are sorted with decreasing order with respect to category distribution in LitBank to further highlight cross-domain transfer performances.

Stage	Annotation
Example 1	
CNER	While all this is going on, Mr. [Clinton]$_{\text{PER}}$ is overseas. [President Clinton]$_{\text{PER}}$ was in [Northern Ireland]$_{\text{LOC}}$ when he heard the [Supreme Court]$_{\text{ORG}}$ [decision]$_{\text{EVENT}}$. He talked to [Al Gore]$_{\text{PER}}$ on the [phone]$_{\text{ARTIFACT}}$ from [Belfast]$_{\text{LOC}}$. This is Mr. [Clinton]$_{\text{PER}}$'s [third]$_{\text{MEASURE}}$ [visit]$_{\text{EVENT}}$ to [Northern Ireland]$_{\text{LOC}}$"
Coreference	"While all this is going on, [Mr. Clinton]$_1$ is overseas. [President Clinton]$_1$ was in [Northern Ireland]$_2$ when [he]$_1$ heard the Supreme Court decision. [He]$_1$ talked to Al Gore on the phone from Belfast. This is [Mr. Clinton]$_1$'s third visit to [Northern Ireland]$_2$"
(1) Mention Assignment	"While all this is going on, [Mr. Clinton]$_{\text{1-PER}}$ is overseas. was in [Northern Ireland]$_{\text{2-LOC}}$ when [he]$_{\text{1-Unassigned}}$ heard the Supreme Court decision . [He]$_{\text{1-Unassigned}}$ talked to Al Gore on the phone from Belfast. This is [Mr. Clinton]$_{\text{1-PER}}$'s third visit to [Northern Ireland]$_{\text{2-LOC}}$"
(2) Cluster Propagation	" While all this is going on, [Mr. Clinton]$_{\text{1-PER}}$ is overseas. [President Clinton]$_{\text{1-PER}}$ was in [Northern Ireland]$_{\text{2-LOC}}$ when [he]$_{\text{1-PER}}$ heard the Supreme Court decision . [He]$_{\text{1-PER}}$ talked to Al Gore on the phone from Belfast. This is [Mr. Clinton]$_{\text{1-PER}}$'s third visit to [Northern Ireland]$_{\text{2-LOC}}$"
Example 2	
CNER	"[Tomorrow]$_{\text{DATETIME}}$'s [summit]$_{\text{EVENT}}$ [meeting]$_{\text{EVENT}}$ will bring [Ehud Barak]$_{\text{PER}}$ and [Yasser Arafat]$_{\text{PER}}$ to the [resort city]$_{\text{STRUCT}}$ of [Sharm El - eikh]$_{\text{LOC}}$. Getting both to attend was not an easy task."
Coreference	"Tomorrow's summit meeting will bring [[Ehud Barak]$_1$ and [Yasser Arafat]$_2$]$_3$ to the resort city of Sharm El-eikh. Getting [both]$_3$ to attend was not an easy task."
(1) Mention Assignment	"Tomorrow's summit meeting will bring [[Ehud Barak]$_{\text{1-PER}}$ and [Yasser Arafat]$_{\text{2-PER}}$]$_{\text{3-PER}}$ to the resort city of Sharm El-eikh. Getting [both]$_{\text{3-Unassigned}}$ to attend was not an easy task."
(2) Cluster Propagation	"Tomorrow's summit meeting will bring [[Ehud Barak]$_{\text{1-PER}}$ and [Yasser Arafat]$_{\text{2-PER}}$]$_{\text{3-PER}}$ to the resort city of Sharm El-eikh. Getting [both]$_{\text{3-PER}}$ to attend was not an easy task."

Table 6.9. Propagation examples annotation

6.5.3 Qualitative Evaluation

CNER propagation examples. In Table 6.9, we show two qualitative examples of CNER propagation heuristic.

In the first example, from a news-domain text in OntoNotes, CNER label propagation ensures type consistency across all mentions within the same entity cluster. In the first row, we show the high coverage of named entities and concepts as CNER annotated spans, while in the second row, we report the different span

boundaries found in the gold annotated coreferences of OntoNotes. During the Mention Assignment stage, we can assign to every non-pronoun mention a CNER label, and with cluster propagation, we can obtain complete coverage of coreference mentions.

The same happens in the second example, in which we can propagate our CNER labels for plural references. In fact, CNER annotation assigns PER to the named individuals Ehud Barak and Yasser Arafat, and the Mention Assignment step tags them correctly as PER, leaving out plural references, which are inherited in the following Cluster Propagation step, demonstrating the heuristic's ability to extend semantic categories across implicitly linked mentions.

Cross-domain Examples. Table 6.10 presents qualitative examples illustrating cross-domain discrepancies among our MAVERICK-MES models trained on LitBank, OntoNotes, and PreCo. These examples highlight systematic differences in mention boundary detection, semantic category coverage, and cross-type linking behavior.

In the first two examples, we observe mention extraction failures in the LitBank-trained model, particularly for underrepresented categories such as CULTURE and DISEASE. In Example 1, the LitBank model identifies only generic *Group* mentions (e.g., "people," "some," "others"), whereas conceptual entities like *the movement* remain unlabeled. In contrast, the OntoNotes model correctly extracts these and assigns CULTURE and MEDIA categories, reflecting broader semantic coverage.

In Example 2, the LitBank model omits all disease-related mentions, while OntoNotes successfully labels *malaria* and related expressions, as well as biological entities such as *the insects*. This underscores OntoNotes' stronger lexical diversity and its inclusion of scientific and factual content, compared to LitBank's narrative focus.

Examples 3 and 4 highlight linking errors involving financial entities. Even with gold mentions, the LitBank model fails to link money-related expressions, treating them as singletons, whereas the PreCo model clusters them accurately and correctly handles possessive references (e.g., *my money*).

Finally, Example 5 demonstrates that the LitBank-trained model fails to link

Model	Output
Ex. 1	*maverick-mes-litbank does not recognize CULTURE mentions.*
maverick-litbank	"... And [people]$_{\text{singleton-GROUP}}$ have different opinions about the movement. [Some]$_{\text{singleton-GROUP}}$ think street art is a crime and destroys property. But [others]$_{\text{singleton-GROUP}}$ see this art as a rich form of non-traditional cultural expression. [Many experts]$_{\text{singleton-GROUP}}$ say the movement began in [New York City]$_{\text{1-LOC}}$ in the nineteen sixties..."
maverick-ontonotes	... And people have different opinions about [the movement]$_{\text{2-CULTURE}}$. Some think [street art]$_{\text{3-MEDIA}}$ is a crime and destroys property. But others see [this art]$_{\text{3-MEDIA}}$ as a rich form of non-traditional cultural expression. Many experts say [the movement]$_{\text{2-CULTURE}}$ began in [New York City]$_{\text{4-LOC}}$ in the nineteen sixties..."
Ex. 2	*maverick-mes-litbank does not recognize DISEASE mentions.*
maverick-litbank	"... The insects carry serious diseases like malaria. It is estimated that almost 630,000 people died from malaria and malariarelated causes in 2012, and most of these cases were in African countries..."
maverick-ontonotes	"... [The insects]$_{\text{0-BIOLOGY}}$ carry serious diseases like [malaria]$_{\text{1-DISEASE}}$. It is estimated that almost 630,000 people died from [malaria]$_{\text{1-DISEASE}}$ and [malariarelated]$_{\text{1-DISEASE}}$ causes in 2012, and most of these cases were in African countries..."
Ex. 3	*Using gold mentions, maverick-mes-litbank is unable to cluster MONEY*
maverick-litbank	"... [With the previous 800 yuan income tax threshold]$_{\text{singleton-MONEY}}$, if [it]$_{\text{singleton-Unassigned}}$ were strictly enforced, 80% of beggars would have to pay..."
maverick-preco	" ... With [the previous 800 yuan income tax threshold]$_{\text{6-MONEY}}$, if [it]$_{\text{6-MONEY}}$ were strictly enforced, 80% of beggars would have to pay..."
Ex. 4	*Using gold mentions, maverick-mes-litbank is unable to cluster MONEY*
maverick-litbank	"...When [he]$_{\text{7-PER}}$ came home, [he]$_{\text{7-PER}}$ said, 'Call [those servants who have [my]$_{\text{7-PER}}$ money]$_{\text{10-PER}}$...'"
maverick-preco	"...When [he]$_{\text{8-PER}}$ came home, [he]$_{\text{8-PER}}$ said, 'Call [those servants who have [[my]$_{\text{8-PER}}$ money]$_{\text{15-MONEY}}$]$_{\text{10-PER}}$...'"
Ex. 5	*Using gold mentions, maverick-mes-libank is unable to cluster DISEASE*
maverick-litbank	"...[Sharon Osbourne, [Ozzy's]$_{\text{0-PER}}$ long-time manager, wife and best friend]$_{\text{1-PER}}$, announced to the world that [she]$_{\text{1-PER}}$ 'd been diagnosed with colon cancer. ... [Sharon]$_{\text{1-PER}}$ announced in April that the cancer was in remission, but just weeks after that announcement, [son Jack]$_{\text{5-PER}}$ entered drug and alcohol rehabilitation in California..."
maverick-preco	"...[Sharon Osbourne, [Ozzy's]$_{\text{0-PER}}$ long-time manager, wife and best friend]$_{\text{1-PER}}$, announced to the world that [she]$_{\text{1-PER}}$ 'd been diagnosed with [colon cancer]$_{\text{3-DISEASE}}$. ... [Sharon]$_{\text{1-PER}}$ announced in April that [the cancer]$_{\text{3-DISEASE}}$ was in remission, but just weeks after that announcement, [son Jack]$_{\text{4-PER}}$ entered drug and alcohol rehabilitation in California..."

Table 6.10. Representative cross-domain annotation errors. Each example contrasts the LitBank-trained model with OntoNotes or PreCo models, showing domain-specific gaps in semantic coverage and mention linking.

disease mentions (*colon cancer*, *the cancer*), due to the scarcity of such categories in LitBank. The PreCo model, by contrast, clusters them consistently.

Overall, these qualitative examples reveal that LitBank-trained models struggle with non-human and abstract referents, such as cultural, financial, and biomedical entities, due to LitBank's narrative bias and limited exposure to factual or technical domains. This domain imbalance results in reduced mention recall and weaker cross-type linking generalization.

6.6 Summary and Future Works

In this chapter, we introduced CNER, an extension of the traditional NER framework that unifies named entities and nominal concepts under a shared inventory of semantic types. We then presented a simple yet effective overlap-based heuristic to propagate CNER annotations to coreference clusters, enabling high-coverage semantic typing of coreference model outputs.

By integrating semantic labels through CNER, we make it possible to perform interpretable, class-aware evaluations of coreference resolution models. Our analysis demonstrates that cross-domain transfer remains limited, not only due to differences in annotation guidelines, which significantly influence mention extraction and evaluation, but also because of uneven coverage of entity types resulting from inconsistent annotation policies across datasets.

We believe this work provides a foundation for future research aimed at expanding the semantic coverage of training corpora and at developing complementary tools for interpretable evaluation for coreference resolution.

Chapter 7

Conclusions and Future Work

Throughout this thesis, we have shown that, despite its central role, research in Coreference Resolution has historically developed under narrow conditions, dealing mainly with short documents in limited domains, leaving open the question of how to make CR both efficient and high-performing at scale.

The work presented in this thesis set out to address these challenges and advance the state of the art in coreference resolution along four overarching axes: improving the efficiency of CR models, developing resources that extend the length to very long documents, proposing a unified modeling framework capable of operating across different text granularities, and enhancing evaluation through semantically-grounded interpretability. This thesis advances these goals through a series of interconnected contributions, each building upon the previous and addressing key limitations in existing work.

The resulting framework advances from efficient modeling to large-scale discourse coverage, from efficient processing of individual documents to enabling unified cross-document reasoning, and from purely statistical evaluation to semantically enriched, interpretable methods. Together, these contributions enable coreference resolution to be modeled, evaluated, and applied across diverse domains and document lengths. By simultaneously achieving efficiency, scalability, and interpretability, this work aims to bridge many of the gaps between benchmark-driven research and the practical deployment of CR in real-world scenarios. **Combining efficiency,**

performance, and usability. Our first contribution, Maverick, establishes a new optimal balance between computational efficiency and coreference resolution modeling capabilities. Defying recent trends toward large generative architectures, Maverick demonstrates that an encoder-only design, when carefully optimized for span representation and inference efficiency, can achieve state-of-the-art performance on benchmarks such as OntoNotes. Beyond raw performance, Maverick emphasizes usability and reproducibility under realistic resource constraints. Its growing adoption in subsequent research attests to its impact, making coreference resolution more approachable, scalable, and usable as a component of downstream NLP systems.

From short documents to full-length discourse. Building on this foundation, the second contribution, BookCoref, addressed one of the most persistent limitations in CR research: the absence of reliable long-document benchmarks. By constructing the first large-scale corpus of full-length books annotated for coreference, comprising 50 automatically tagged and 3 manually annotated narrative texts, BookCoref provided the empirical ground needed to study coreference resolution across hundreds of pages. Our automatic annotation mechanism was specifically designed to leverage Maverick's high performance and efficiency, enabling the application of coreference resolution to long contexts. In BookCoref, our thorough analysis of current methods interestingly revealed that models trained on short or medium-length texts, despite strong benchmark performance, are usually not applicable or fail to preserve coherence in very long contexts. **Unifying short-, long- and cross-document coreference.** Our third contribution, xCoRe, extended the principles of efficiency and scalability established in Maverick and the long-document insights from BookCoref to propose the Cross-context Coreference setting, a first unified framework for short-, long-, and cross-document CR, and introduced xCoRe, a new unified approach designed to robustly handle those scenarios. xCoRe, adopting a three-step pipeline that first identifies mentions, then creates clusters within individual contexts, and finally merges clusters across contexts, marked a conceptual shift: instead of treating short, long, and multi-document CR as separate tasks, xCoRe approached them jointly, achieving state-of-the-art results across multiple benchmarks. Following the principles established in Maverick, xCoRe also prioritizes computational feasibility,

showing that high performance can be reached without sacrificing efficiency. In doing so, it consolidates the architectural and efficiency advances of the previous contributions into a unified, practical modeling paradigm.

Semantically grounded evaluation. Finally, our fourth contribution addresses the persistent challenge of interpretability in Coreference Resolution evaluation. Traditional statistical metrics, while widely adopted, provide limited insight into the semantic nature of model errors or the impact of domain and annotation shifts. By integrating Concept and Named Entity Recognition (CNER) into the coreference evaluation pipeline, this work introduced a fine-grained and robust way to assign semantic categories to coreference clusters. This enabled class-specific analyses of model behavior, revealing that cross-domain degradation is often linked not only to annotation mismatches but also to uneven coverage of semantic types. Through this semantically grounded evaluation, we were able to uncover systematic biases in model performance and provide a more interpretable, class-aware understanding of CR systems.

Taken together, these four contributions are tightly interconnected to advance automatic Coreference Resolution: **Maverick** establishes efficiency and usability; **BookCoref** extends CR to realistic discourse lengths; **xCoRe** unifies short-, long-, and cross-document modeling; and leveraging **CNER**, we can enhance evaluation with semantic interpretation. Collectively, these contributions expand CR from efficient modeling to comprehensive discourse understanding and interpretable evaluation.

7.1 Future Directions

While the contributions of this work advance the field in multiple directions, they also open rich avenues for future exploration.
Several research directions naturally emerge:

- **Multilinguality and multimodality.** Extending CR beyond English remains an open challenge and a key step toward truly universal discourse modeling.

Current benchmarks and models are predominantly in English, which limits generalization to low-resource or typologically distant languages. A natural extension of our work is to adapt our proposed architectures to multilingual text. Leveraging multilingual pre-trained transformers could enable effective transfer of CR capabilities to underrepresented languages beyond English with minimal effort in terms of architecture modifications; however, solving the lack of high-quality multilingual data should be prioritized. Beyond textual modalities, the growing availability of vision-text datasets opens the frontier of multimodal coreference. Future systems could incorporate visual grounding, linking mentions not only to textual entities but also to visual referents in images or videos, paving the way for unified discourse understanding across modalities.

- **Coreference in downstream reasoning and generation tasks.** Coreference Resolution plays a critical but often implicit role in many downstream NLP applications such as summarization, dialogue, factual consistency checking, and question answering. A key direction for future research is to develop systems that integrate CR dynamically within generative or reasoning pipelines. For example, maintaining and updating entity states through CR could help large language models produce more coherent and factually consistent long-form reasoning, which is currently a very important frontier to improve those models' capabilities: Embedding CR-aware reasoning into retrieval-augmented generation or chain-of-thought models could provide explicit interpretability of entity references across generated sequences. Finally, tight integration between CR and external knowledge bases could enable entity linking and reference resolution to reinforce one another, strengthening the factual grounding of neural language systems.

- **Improving modeling and evaluation of long- and cross-document settings.** Despite substantial progress with BookCoref and xCoRe, the challenges of scaling CR to entire books or collections of related documents remain formidable. Future research could pursue several directions:

1. *Metric reinterpretation and development:* In our work, we found that existing coreference metric scores have very low agreement when measuring performance on very long coreference chains in full books. Investigating how existing coreference metrics behave when applied to book-length or cross-document corpora and how to correctly interpret those scores is crucial for comparing short-document to long-document results. Furthermore, designing new evaluation metrics that better capture entity continuity and narrative coherence over long spans would be crucial to enable correct measurements in this new scenario.
2. *Efficient long-context adaptation:* Exploring how to correctly and efficiently adapt generative models is crucial for their application to this new book-scale setting. Some directions could experiment using hierarchical and memory-augmented models that combine local mention detection with global discourse tracking. Some inspiration can be found from retrieval-based or sparse attention mechanisms to handle large contexts efficiently in different tasks, such as Summarization or Question Answering.

While xCoRe offers an initial step toward scalable cross-document CR, achieving real-time or streaming-level CR for large collections of documents remains an open and impactful goal.

- **Enhancing the interplay between semantics and coreference.** Finally, one of the most promising future directions lies in deepening the bidirectional relationship between semantics and coreference. In this work, we introduced a simple mechanism for semantic propagation within the CNER framework; however, many richer interactions remain unexplored. Future efforts could focus on:
  - *Semantic feedback to coreference:* Improve the usage of explicit semantic cues, such as entity types, conceptual hierarchies, or contextual embeddings, to prevent incorrect clustering of semantically incompatible mentions, improving both precision and interpretability.
  - *Coreference feedback to semantics:* Leverage CR to refine CNER pre-

dictions by propagating type or conceptual information across mentions within a cluster, enhancing consistency across full contexts.

– *Joint learning frameworks:* Develop multitask or co-training approaches where CR and semantic classification inform each other during training, promoting shared representations that capture both discourse structure and meaning.

A tighter coupling between these two perspectives could yield models that not only resolve references but also understand them, moving CR closer to genuine discourse-level comprehension.

7.2 Final Remarks

The work presented in this dissertation demonstrates that progress in Coreference Resolution does not depend only on scaling data or model size, but on fundamentally rethinking the problem: how models represent and connect entity mentions, how benchmarks can capture challenging long-text capabilities, and how evaluation reflects meaning. In this sense, the path forward for Coreference Resolution is also the path forward for Natural Language Understanding: moving from sentence-level recognition to document-level comprehension, from static benchmarks to dynamic reasoning, and from opaque outputs to interpretable understanding of entities and their relationships.

Bibliography

Dhruv Agarwal, Rico Angell, Nicholas Monath, and Andrew McCallum. 2022. Entity linking via explicit mention-mention coreference modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4644–4658, Seattle, United States. Association for Computational Linguistics.

META AI. 2024. The llama 3 herd of models.

Gerry Altmann and Mark Steedman. 1988a. Interaction with context during human sentence processing. *Cognition*, 30(3):191–238.

Gerry Altmann and Mark Steedman. 1988b. Interaction with context during human sentence processing. *Cognition*, 30(3):191–238.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 79–85, Montreal, Quebec, Canada. Association for Computational Linguistics.

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.

David Bamman, Brendan O'Connor, and Noah A. Smith. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), pages 352–361, Sofia, Bulgaria. Association for Computational Linguistics.

Edoardo Barba, Tommaso Pasini, and Roberto Navigli. 2021. ESC: Redesigning WSD with extractive sense comprehension. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4661–4672, Online. Association for Computational Linguistics.

Sabyasachee Baruah, Sandeep Nallan Chakravarthula, and Shrikanth Narayanan. 2021. Annotation and evaluation of coreference resolution in screenplays. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2004–2010, Online. 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*, pages 10300–10313, Toronto, Canada. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020a. Longformer: The long-document transformer.

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

Bernd Bohnet, Chris Alberti, and Michael Collins. 2023. Coreference resolution through a seq2seq transition-based system. *Transactions of the Association for Computational Linguistics*, 11:212–226.

Mariya Borovikova, Loïc Grobol, Anaïs Halftermeyer, and Sylvie Billot. 2022. A methodology for the comparison of human judgments with metrics for coreference resolution. In *Proceedings of the 2nd Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 16–23, Dublin, Ireland. Association for Computational Linguistics.

Faeze Brahman, Meng Huang, Oyvind Tafjord, Chao Zhao, Mrinmaya Sachan, and Snigdha Chaturvedi. 2021. "let your characters tell their story": A dataset for character-centric narrative understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1734–1752, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Haixia Chai, Nafise Sadat Moosavi, Iryna Gurevych, and Michael Strube. 2022. Evaluating coreference resolvers on community-based question answering: From rule-based to state of the art. In *Proceedings of the Fifth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 61–73, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Henry Y. Chen, Ethan Zhou, and Jinho D. Choi. 2017. Robust coreference resolution and entity linking on dialogues: Character identification on TV show transcripts. In *Proceedings of the 21st Conference on Computational Natural Language Learning*

(CoNLL 2017), pages 216–225, Vancouver, Canada. Association for Computational Linguistics.

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.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96, Melbourne, Australia. Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2015. Entity-centric coreference resolution with model stacking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1405–1415, Beijing, China. 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).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Vladimir Dobrovolskii. 2021. Word-level coreference resolution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*,

pages 7670–7675, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490.

Tobias Falke, Christian M. Meyer, and Iryna Gurevych. 2017. Concept-map-based multi-document summarization using concept coreference resolution and global importance optimization. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 801–811, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Yujian Gan, Yuan Liang, Yanni Lin, Juntao Yu, and Massimo Poesio. 2025. Improving LLMs' learning of coreference resolution. In *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 311–321, Avignon, France. Association for Computational Linguistics.

Yujian Gan, Massimo Poesio, and Juntao Yu. 2024. Assessing the capabilities of large language models in coreference: An evaluation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1645–1665, Torino, Italia. ELRA and ICCL.

Abbas Ghaddar and Phillippe Langlais. 2016. WikiCoref: An English coreference-annotated corpus of Wikipedia articles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 136–142, Portorož, Slovenia. European Language Resources Association (ELRA).

Venkata S Govindarajan, Matianyu Zang, Kyle Mahowald, David Beaver, and Junyi Jessy Li. 2024. Do *they* mean 'us'? interpreting referring expression variation under intergroup bias. In *Findings of the Association for Computational*

Linguistics: EMNLP 2024, pages 9772–9785, Miami, Florida, USA. Association for Computational Linguistics.

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.

Yifan Guo, Hongying Zan, and Hongfei Xu. 2024. Dual-teacher knowledge distillation for low-frequency word translation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5543–5552, Miami, Florida, USA. 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.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

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.

Youngjoon Jang, Seongtae Hong, Junyoung Son, Sungjin Park, Chanjun Park, and Heuiseok Lim. 2025. From ambiguity to accuracy: The transformative effect of coreference resolution on retrieval-augmented generation systems. In *Proceedings of*

the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 422–433, Vienna, Austria. 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.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5803–5808, Hong Kong, China. Association for Computational Linguistics.

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

Sopan Khosla and Carolyn Rose. 2020. Using type information to improve entity coreference resolution. In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pages 20–31, Online. 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 (Volume 2: Short Papers)*, pages 14–19, Online. Association for Computational Linguistics.

Vincent Labatut and Xavier Bost. 2019. Extraction and Analysis of Fictional Character Networks: A Survey. *ACM Comput. Surv.*, 52(5).

Nghia T. Le and Alan Ritter. 2024. Are Language Models Robust Coreference Resolvers? In *First Conference on Language Modeling*.

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.

Maolin Li, Hiroya Takamura, and Sophia Ananiadou. 2020. A neural model for aggregating coreference annotation in crowdsourcing. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5760–5773, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Tianyu Liu, Yuchen Eleanor Jiang, Nicholas Monath, Ryan Cotterell, and Mrinmaya Sachan. 2022. Autoregressive structured prediction with language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 993–1005, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Yanming Liu, Xinyue Peng, Jiannan Cao, Yanxin Shen, Tianyu Du, Sheng Cheng, Xun Wang, Jianwei Yin, and Xuhong Zhang. 2024. Bridging context gaps: Leveraging coreference resolution for long contextual understanding. *arXiv preprint arXiv:2410.01671*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhengyuan Liu, Ke Shi, and Nancy Chen. 2021. Coreference-aware dialogue summarization. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 509–519, Singapore and Online. Association for Computational Linguistics.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Xiaoqiang Luo, Sameer Pradhan, Marta Recasens, and Eduard Hovy. 2014. An extension of BLANC to system mentions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 24–29, Baltimore, Maryland. Association for Computational Linguistics.

Kawshik S. Manikantan, Shubham Toshniwal, Makarand Tapaswi, and Vineet Gandhi. 2024. Major entity identification: A generalizable alternative to coreference resolution. In *Proceedings of the Seventh Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 1–17, Miami. Association for Computational Linguistics.

Giuliano Martinelli, Edoardo Barba, and Roberto Navigli. 2024a. 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.

Giuliano Martinelli, Francesco Molfese, Simone Tedeschi, Alberte Fernández-Castro, and Roberto Navigli. 2024b. CNER: Concept and named entity recognition. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8336–8351, Mexico City, Mexico. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

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.

Frank Mtumbuka and Steven Schockaert. 2024. EnCore: Fine-grained entity typing by pre-training entity encoders on coreference chains. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1768–1781, St. Julian's, Malta. Association for Computational Linguistics.

Roberto Navigli, Michele Bevilacqua, Simone Conia, Dario Montagnini, and Francesco Cecconi. 2021. Ten years of BabelNet: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4559–4567. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Vincent Ng. 2007. Shallow semantics for coreference resolution. In *IJcAI*, volume 7, pages 1689–1694.

Takumi Ohtani, Hidetaka Kamigaito, Masaaki Nagata, and Manabu Okumura. 2019. Context-aware neural machine translation with coreference information. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 45–50, Hong Kong, China. Association for Computational Linguistics.

Riccardo Orlando, Pere-Lluís Huguet Cabot, Edoardo Barba, and Roberto Navigli. 2024. ReLiK: Retrieve and LinK, fast and accurate entity linking and relation extraction on an academic budget. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14114–14132, Bangkok, Thailand. Association for Computational Linguistics.

Shon Otmazgin, Arie Cattan, and Yoav Goldberg. 2022. F-coref: Fast, accurate and easy to use coreference resolution. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 48–56, Taipei, Taiwan. 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.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. *arXiv preprint arXiv:2101.05779*.

Ramakanth Pasunuru, Mengwen Liu, Mohit Bansal, Sujith Ravi, and Markus Dreyer. 2021. Efficiently summarizing text and graph encodings of multi-document clusters.

In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4768–4779, Online. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Ian Porada and Jackie CK Cheung. 2024. Solving the challenge set without solving the task: On Winograd schemas as a test of pronominal coreference resolution. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pages 489–506, Miami, FL, USA. Association for Computational Linguistics.

Ian Porada, Alexandra Olteanu, Kaheer Suleman, Adam Trischler, and Jackie Cheung. 2024a. Challenges to evaluating the generalization of coreference resolution models: A measurement modeling perspective. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15380–15395, Bangkok, Thailand. Association for Computational Linguistics.

Ian Porada, Xiyuan Zou, and Jackie Chi Kit Cheung. 2024b. 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, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coref-

erence in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Ina Roesiger, Sarah Schulz, and Nils Reiter. 2018. Towards coreference for literary text: Analyzing domain-specific phenomena. In *Proceedings of the Second Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 129–138, Santa Fe, New Mexico. Association for Computational Linguistics.

Henry Rosales-Méndez, Aidan Hogan, and Barbara Poblete. 2020. Fine-grained entity linking. *Journal of Web Semantics*, 65:100600.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Kumar Shridhar, Nicholas Monath, Raghuveer Thirukovalluru, Alessandro Stolfo, Manzil Zaheer, Andrew McCallum, and Mrinmaya Sachan. 2023. Longtonotes: OntoNotes with longer coreference chains. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1428–1442, Dubrovnik, Croatia. Association for Computational Linguistics.

Dario Stojanovski and Alexander Fraser. 2018. Coreference and coherence in neural machine translation: A study using oracle experiments. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 49–60, Brussels, Belgium. 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.

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*, pages 769–774, Lisbon, Portugal. Association for Computational Linguistics.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Kellie Webster and James R. Curran. 2014. Limited memory incremental coreference resolution. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2129–2139, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the GAP: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.

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, Mariama Drame, Quentin Lhoest, and Alexander Rush. 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.

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

Patrick Xia, João Sedoc, and Benjamin Van Durme. 2020. Incremental neural coreference resolution in constant memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8617–8624, Online. Association for Computational Linguistics.

Yiyun Xiong, Mengwei Dai, Fei Li, Hao Fei, Bobo Li, Shengqiong Wu, Donghong Ji, and Chong Teng. 2023. Dialogre c+: An extension of dialogre to investigate how much coreference helps relation extraction in dialogs. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 222–234. Springer.

Fangyuan Xu, Junyi Jessy Li, and Eunsol Choi. 2022. How do we answer complex questions: Discourse structure of long-form answers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3556–3572, Dublin, Ireland. Association for Computational Linguistics.

Huanwei Xu, Lin Xu, and Liang Yuan. 2025. Clap: Coreference-linked augmentation for passage retrieval. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 3612–3622.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue,

Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report.

Asaf Yehudai, Arie Cattan, Omri Abend, and Gabriel Stanovsky. 2023. Evaluating and improving the coreference capabilities of machine translation models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 980–992, Dubrovnik, Croatia. Association for Computational Linguistics.

Klim Zaporojets, Johannes Deleu, Yiwei Jiang, Thomas Demeester, and Chris Develder. 2022. Towards consistent document-level entity linking: Joint models for entity linking and coreference resolution. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 778–784, Dublin, Ireland. Association for Computational Linguistics.

Daojian Zeng, Chao Zhao, Chao Jiang, Jianling Zhu, and Jianhua Dai. 2023. Document-level relation extraction with context guided mention integration and inter-pair reasoning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Rui Zhang, Cícero Nogueira dos Santos, Michihiro Yasunaga, Bing Xiang, and Dragomir Radev. 2018. Neural coreference resolution with deep biaffine attention by joint mention detection and mention clustering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 102–107, Melbourne, Australia. 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. LlmLink: 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.