CorefInst: Leveraging LLMs for Multilingual Coreference Resolution

Tuğba Pamay Arslan◇ and **Emircan Erol**◇ and **Gülşen Eryiğit**◇

◇İTÜ NLP Research Group, Turkey
Department of Artificial Intelligence and Data Engineering
İstanbul Technical University, Turkey
[pamay, erole20, gulsen.cebiroglu]@itu.edu.tr

Abstract

Coreference Resolution (CR) is a crucial yet challenging task in natural language understanding, often constrained by task-specific architectures and encoder-based language models that demand extensive training and lack adaptability. This study introduces the first multilingual CR methodology which leverages decoder-only LLMs to handle both overt and zero mentions. The article explores how to model the CR task for LLMs via five different instruction sets using a controlled inference method. The approach is evaluated across three LLMs: Llama 3.1, Gemma 2, and Mistral 0.3. The results indicate that LLMs, when instruction-tuned with a suitable instruction set, can surpass state-of-the-art task-specific architectures. Specifically, our best model, a fully fine-tuned Llama 3.1 for multilingual CR, outperforms the leading multilingual CR model (i.e., Corpipe 24 single stage variant) by 2 percentage points on average across all languages in the CorefUD v1.2 dataset collection.

1 Introduction

Coreference Resolution (CR) is a semantic-level task in natural language processing (NLP) that involves detecting and clustering mentions referring to the same entity within a text. An end-to-end CR system consists of two phases: mention detection and coreference linking. In mention detection, all possible referential mentions are extracted from the text. In the next step, mentions referring to the same entity are collected under the same cluster, by resolving which mentions are coreferential. This task is crucial for various higher-level NLP applications, including machine translation (Stojanovski and Fraser, 2018), text summarization (Liu et al., 2021), and question answering systems (Bhattacharjee et al., 2020), as it helps in understanding the coherence and context of the text.

Large Language Models (LLMs) are traditionally classified into three categories: encoder-only models (e.g., BERT), encoder-decoder models (e.g., T5), and decoder-only models (e.g., GPT, Llama). Initial task-specific neural CR models (Lee et al., 2017; Straka, 2023) incorporate an encoder-only LLM at the bottom layer to generate word embeddings, followed by a specialized neural architecture where the LLM layer is also fine-tuned during the training process, illustrated in Figure 1. However, task-specific models require specialized architectures, vast amount of training data, and an extensive fine-tuning/training process from scratch, leading to limited generalizability and flexibility across new datasets, languages, tasks, and domains. Furthermore, due to their niche nature, these models cannot exploit decoder-based LLMs directly, which have been rapidly developing. On the other hand, instruction tuning has emerged as a powerful paradigm, enabling decoder-only LLMs to adapt to diverse tasks through carefully crafted prompts or instructions, thereby addressing these limitations. It is important to note that recently, LLMs are often used interchangeably with decoder-only models in the literature, particularly due to the popularity of autoregressive models like GPT. From this point onward in the article, we will refer to decoder-only LLMs simply as LLMs.

Despite notable progress in CR, the technologies developed to tackle challenges such as handling zero mentions and generalizing across various languages are still evolving (Novák et al., 2024; Pamay Arslan and Eryiğit, 2025). Recent CR studies (Chen et al., 2021; Straka, 2023; Pamay Arslan and Eryiğit, 2025) address zero mentions and their coreferential relations alongside overt mentions. Zero mentions are not explicitly defined in an original sentence, and result from dropped-pronouns in pro-dropped languages (e.g., Chinese, Hungarians, Turkish). Figure 2 presents
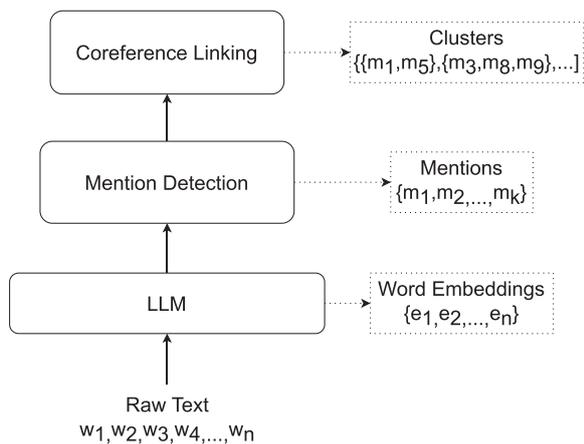
Figure 1: A sample flow of a traditional CR-specific neural architecture.

| (on) | Odešel, | ale | (on) | vrátil | se. |
| He | left | but | he | came | back. |

He left but came back.

Figure 2: Zero mention in pro-drop language example.

a sample Czech sentence containing zero mentions along with its English translation. As seen, the zero pronoun, 'on' (Eng. he), is not explicitly stated in the Czech sentence; in other words, it is omitted. CR systems considering zeros also incorporate these latent coreferential relations arising from dropped pronouns.

To explore the effectiveness and potential of LLMs in multilingual CR, this study seeks answers to the following main research questions:

- How can LLMs be effectively leveraged for multilingual CR, including the resolution of both overt and zero mentions?

- Can instruction-tuned LLMs outperform traditional, task-specific architectures for multilingual CR?

- How well do instruction-tuned LLMs resolve coreferences arising from zero mentions?

To address these questions, three state-of-the-art LLMs (Llama 3.1, Gemma-2, and Mistral 0.3) were initially analyzed using five different instruction sets to provide preliminary insights into their CR capabilities. Based on these initial evaluations, the most effective LLM-instruction set pair was selected for further analysis. The se-

lected model, hereafter referred to as CorefInst, was first examined under full training and few-shot settings, and then evaluated in an end-to-end manner on predicted mentions (both overt and zero) and compared against CorPipe24-single (Straka, 2024) on the CorefUD dataset v1.2 (Popel et al., 2024), yielding better performances. Finally, the capability of CorefInst[1] to resolve zero mentions was specifically assessed, demonstrating superior performance on pro-drop languages.

The major contributions of the article are: 1) the first instruction-tuned LLM for multilingual CR, and 2) a controlled inference method to model the CR task for LLMs. The proposed methodology for instruction-based fine-tuning of LLMs as a coreference resolver also incorporates zero mentions in addition to overt ones. The article is structured as follows: Section 2 introduces the available studies in the literature, Section 3 presents our proposed model in detail, Section 4 gives the experimental settings and results, and Section 5 concludes the article.

2 Related Work

Early neural CR studies typically employ task-specific neural architectures which utilize an encoder component of a large language model (i.e., encoder-only or encoder-decoder LLMs) to generate word embeddings. Initially, these approaches exclusively focus on the coreference linking stage (Clark and Manning, 2016; Wiseman et al., 2016), but in recent years, end-to-end methods, where mention detection and coreference linking stages are trained either jointly or sequentially, have become a dominant approach in the CR literature (Lee et al., 2018; Kantor and Globerson, 2019; Joshi et al., 2020; D'Oosterlinck et al., 2023; Straka, 2024; Liu et al., 2024).

With the increasing popularity of generative LLMs which contain a decoder component (e.g., T0 [Sanh et al., 2022], Llama [Touvron et al., 2023], GPT [Yenduri et al., 2024]), adapting them to NLP tasks using zero-/few-shot prompting or learning techniques has recently emerged as a prominent research strategy. However, CR studies in this context are still in their early stages. Brown et al. (2020) demonstrated that prompting GPT-3 can resolve coreferential relations. Yang et al. (2022) propose a question-answering-based

---

[1]The source code is available at this URL.

65

prompt-engineering approach to assess the capabilities and limitations of generative, pre-trained GPT-variant LLMs in the CR, demonstrating that these models perform poorly on the task without fine-tuning. Sanh et al. (2022) introduce T0, a zero-shot generalization of T5 (Raffel et al., 2020), which transforms various supervised datasets into task-specific prompts, including those for CR. Gan et al. (2024) evaluate various LLMs for their CR abilities using zero- and few-shot prompting, concluding that LLMs still have potential for improvement in their CR performances. Le and Ritter (2024) present a prompt-based CR system by employing instruction-tuned LLMs (e.g., InstructGPT, Llama-2-Chat). Besides prompt-based approaches, sequence-to-sequence CR models are also seen in the CR literature (Zhang et al., 2023; Liu et al., 2022; Bohnet et al., 2023; Skachkova, 2024).

The CR studies mentioned up to this point have mainly been developed for English, largely due to the availability of a standardized annotated dataset (Pradhan et al., 2012). However, this dataset is not publicly available due to licensing restrictions. Meanwhile, multilingual CR studies (Skachkova et al., 2023; Pamay Arslan et al., 2023; Straka, 2024) are advancing rapidly in this field, particularly due to the CorefUD initiative and shared tasks conducted in recent years (Žabokrtský et al., 2023; Novák et al., 2024). CorefUD (Nedoluzhko et al., 2022) is an initiative to collect coreference corpora in various languages and harmonize them to the same scheme and data format (CoNLL-U) (De Marneffe et al., 2021). The latest version, CorefUD v1.2 (Popel et al., 2024), consists of 21 datasets for 15 languages, such as English, Russian, French, Norwegian and so on. Additionally, the CorefUD, which also includes pro-drop languages (e.g., Turkish and Catalan) has facilitated the development of CR literature for these languages.

3 The Proposed Method

This study proposes a methodology for instruction-based fine-tuning of an LLM for the multilingual CR task. This section presents the entire instruction engineering process, all the necessary steps for data processing and mention clustering, and the proposed controlled inference method. One should note that the instruction and input/output format selection are inter-dependent

steps. That is, a recursive search combined with a feedback loop method is employed to refine and finalize their formats.

3.1 Instruction Engineering

The proposed method accepts a tuple as an input consisting of three parts: <an instruction, masked input sentences, output sentences>, exemplified in Figure 3. Basically, the model learns to generate the expected output by determining which coreferences are present in masked input sentences, taking the given instruction into consideration. An instruction is a directive that provides the necessary preliminary information about a task. This part may contain details about the input, and the expected output format, and also include some restrictions/limitations that should be considered by the LLM when generating outputs.

We explore five different instruction sets, provided in Table 4 in the Appendix section. Each instruction set has its own characteristic, and may encompass a range of information, including language in focus, different task definitions, explanations, and errors identified in the model's generated outputs. For instance, the first instruction set (Inst. #1) is the most detailed one, presenting straightforward statements regarding the task, input/output format, and expectations from the LLM as a coreference resolver. The second instruction set (Inst. #2) employs an alternative task definition by also providing a sample input/output pair. The remaining instruction sets represent various modifications of each other. Notably, the third (Inst. #3) and fifth (Inst. #5) instruction sets incorporate a non-common keyword in CR, 'coherent',[2] along with 'coreferential'. Although 'coherent' is not a commonly used term in CR literature, in this study it is used to encourage the model to consider discourse-level consistency, as coreference resolution is closely linked to textual coherence and is often used as one of its measurable indicators (Barzilay and Lapata, 2008). It should be noted that this usage is experimental and serves as a complementary guide rather than a replacement for traditional terminology. Additionally, the fourth (Inst. #4) and fifth instruction sets provide a semantically related but

---

[2]The term 'coherent' was proposed by ChatGPT during our preliminary experiments aimed at probing the extent of the LLM's knowledge regarding the CR. This observation implies that the LLM may implicitly associate these two concepts within its internal representational framework.

Instruction
For every mention in <m> </m> tags examine if there is any coreferential/coherent mention. If two mentions represent the same entity write the same number instead of MASK after closing mention tag </m>. For example: author and book represent different entities. Do not change anything else other than MASK. *Where you see </z>@ there is a zero mention, which is normally not written but you also need to link them with other mentions.*[*]
Input
<m> Leges-legkedvesebb napom </m>#MASK az volt, <m> mikor </m>#MASK tavaly június 6.-án mentem a <m> T. sziklási Sport Klub </m>#MASK nagyobb tagjaival </z>@MASK és <m> Pista bácsival </m>#MASK <m> az országos váltóbajnokságra </m>#MASK . [MID] Nagyon örültem, mikor <m> Pista bácsi </m>#MASK mondta , _ hogy megyek <m> a versenyre </m>#MASK , attól a naptól fogva rendszeresen edzettem otthon is és az edzésen is.
Output
<m> Leges-legkedvesebb napom </m>#0 az volt, <m> mikor </m>#0 tavaly június 6.-án mentem a <m> T. sziklási Sport Klub </m>#1 nagyobb tagjaival </z>@1 és <m> Pista bácsival <m>#2 <m> az országos váltóbajnokságra </m>#3 . [MID] Nagyon örültem, mikor <m> Pista bácsi </m>#2 mondta, _ hogy megyek <m> a versenyre </m>#3 , attól a naptól fogva rendszeresen edzettem otthon is és az edzésen is.

(Eng.) <m> My favorite day </m>#0 was <m> when </m>#0 I went to <m> the national relay championship </m>#3 on June 6 last year with the older members of the <m> T. sziklási Sport Club </m>#1 and <m> uncle Pista </m>#2. [MID] I was very happy when <m> uncle Pista </m>#2 told me that I was going to <m> the competition </m>#3, from that day on I trained regularly both at home and in training.

Figure 3: A sample tuple containing Instruction #5 (see Table 4 for all explored instructions) with a masked input in Hungarian, and with a corresponding output. The English translation of the output sentence is also provided at the bottom.

non-coreferent mention pair example to illustrate whether a coreferential relation exists (e.g., ''author'' and ''book'' represent different entities), with Inst. #4 containing only the term 'coreferential'. One should note that the keyword 'coherent' and this additional example are included in instruction sets (i.e., #3, #4, and #5) based on the models' generated errors observed during the instruction engineering process. Moreover, each explored instruction set is complemented by an instruction directing the LLM to identify coreferential relations of zero mentions (depicted in italic font in Table 4) as well.

Table 3 presents a sample tuple containing Inst. #5 with a masked input in Hungarian, and with a corresponding output. The first sentence of the instruction set explains how to resolve coreferential relations for each mention, which is encapsulated by specified mention tags. When the instruction contains only the 'coreferential' keyword, it is observed that the model

tends to over-merge clusters (i.e., collecting many non-coreferential mentions into the same cluster). Consequently, an additional keyword (i.e., coherent) is used together with 'coreferential' to ensure that the model pays attention to referential links that depend more on the flow of the discourse rather than on sentence structure or word repetition. In the second sentence, the model learns to assign a predicted cluster number solely to the MASK token of each mention. The third line provides the above-mentioned example. The forth statement in the instruction set enables to guide the LLM to generate outputs that are as close as possible to expected results, that is, a generated output with only updated MASK token with a predicted cluster number. The last one, written in italic font, is an extra explanation to be used for coreference linking for only pro-dropped languages.

During the instruction engineering process, the performance of each different LLM-instruction set pair is examined, and those with the highest

Global	n	n+1	n	
Local	0	1	0	
Input 1	Jake goes to school. [MID] Amy and Jake are active in the lesson.			
Global	n+1 n	n	n+1	n+2
Local	0 1	1	0	2
Input 2	Amy and Jake are active in the lesson. [MID] He wants to be a scientist and Amy a doctor like her mother.			

Figure 4: Merging clusters with overlapping frames.

achievements are chosen for further experiments. These initial analyses help identify high-performing pairs, offering insights into the potential performance of the proposed model using full training and few-shot learning on the entire dataset.

3.2 Data Processing and Mention Clustering

The proposed model is trained and evaluated on CorefUD v1.2 dataset (Popel et al., 2024) collection, which are formatted in CoNLL-U (De Marneffe et al., 2021), which is a standardized column-wise format used for annotating and representing linguistic information at the morphological, syntactic, and semantic levels in NLP. Therefore, a data processing phase is required to make these datasets compatible with the instruction-based fine-tuning process of an LLM. This section presents the pre-processing step on the raw documents to prepare the masked input sentences, and the post-processing step on the generated outputs to re-convert them to CoNLL-U format for evaluation purposes.

The masked input sentence is a pre-processed version of the original sentence, incorporating mention boundary markers (i.e., <m>,</m> for overt mentions, and </z> for zero mentions) and coreference relation placeholders (i.e., MASK), as presented in the second part of Figure 3, named 'Input'. Mentions may be considered as word sequences which are surrounded by <m>, and </m> tags for the initial and final tokens of an overt mention, respectively. Similarly, </z> is inserted at the exact location of a dropped pronoun in a sentence to indicate zero mentions. One should note that the length of zero mentions is always 1,[3] thus, only closing mark is used for these type of mentions. The main aim of the representation[4] should be less complex, easily interpretable by the model and human-readable. Therefore, we use only three boundaries (<m>, </m>, <z>) without any numbers for entity clusters or mentions to reduce complexity of the representation, and consequently, difficulty in post-processing. In addition, the MASK token is used with # and @ symbols to indicate overt and zero mentions, respectively. This helps to distinguish mention types during the post-processing stage (i.e., converting the predicted output of the LLM to the CoNLL-U data format).

As the final stage of the pre-processing, frames are created from the masked input sentences with a pre-defined length of sequences. The final masked input for the LLM is formed by concatenating two consecutive frames with a separator (i.e., [MID]), which helps to preserve the coreferential relation chain among generated outputs for a document. It should be noted that the length of a document, in terms of the number of tokens, may exceed memory limits or maximum segment length of LLMs. The primary motivation behind using frames is to enable the model to better handle long documents by breaking them down into smaller, more manageable segments (called frames).

The post-processor converts the model's generated output into the CoNLL-U format by transferring the predicted cluster numbers into raw documents (i.e., those having no coreference annotations). The flow of this conversion is illustrated in Figure 4. For simplicity, each frame consists of only one sentence, meaning the proposed model makes prediction on two sentences

---

[3]Syntactically, zero mentions are represented as a single node in the dependency tree of a sentence in which they exist, as they are considered to consist of only a single token.

[4]This representation is inspired by existing studies in the CR literature (Wu et al., 2020; Zhang et al., 2023) as well as our initial experiments on various representations.

in a masked input. The figure provides two different inputs (i.e., Input 1 and Input 2) containing overlapping frames: the right-side (i.e., the second after [MID] separator) of the Input 1 is exactly the same as the left-side of the Input 2. The model predicts a cluster number for each mention, which is indicated just above the mention in the table as variable 'local'. Local number is a predicted cluster number generated as the model's output, and is only meaningful within the frames of a single input. However, a global set of cluster numbers is required to collect all coreferential mentions in the same cluster within a document. Therefore, during the post-processing, each mention is also assigned to its global cluster number with the help of overlapping frames. For example, while the first and second occurrences of 'Jake' in the first input have the same local cluster number of 0, the mention 'Amy' is assigned to a different local cluster number of 1. Since this is the first input, each local cluster is directly mapped to a newly created global cluster (i.e., 0→n, and 1→n+1) and the mappings between them are kept for further inputs. Then, the second input is processed. In the second input, there are two occurrences of 'Amy' in the first and second frames. Since the global cluster numbers for the mentions in the first frame are already established, these occurrences of 'Amy' are directly assigned to an available global cluster, n+1. Similarly, since the mention 'He' is coreferential with 'Jake', its global cluster number is also assigned to n. In the second frame, there is a new mention 'her mother'. Since this is a newly referenced mention among all inputs, a new global cluster is created for this mention with number n+2. In this way, all inputs are post-processed sequentially while preserving the coreferential relation chains within a document. The pseudo-code of this process is explained in Algorithm 1.

Initially, a global cluster dictionary, `globCls`, is initialized by the first frame of the first input along with the model's predicted cluster numbers for that corresponding frame (referred to as 'local' in Figure 4). Keys of `globCls`, `mPos`, represent a mention's position within a document (i.e., start and end words' indices). The following input-output pairs are then processed sequentially in the order they appear within a document. `before` and `after` are lists of mentions that exist in the first and second frames of an input, respectively. The local cluster numbers and word indices of these mentions are stored as attributes

---

Algorithm 1 Merge Clusters over Frames.

1: `globCls` is initialized.
2: **for all** $before, after \in frameTuples$ **do**
3:     $map = \{\}$
4:     **for** $mPos, localNo \leftarrow before$ **do**
5:         $map[localNo] \leftarrow globCls[mPos]$
6:     **end for**
7:     **for all** $mPos, localNo \in after$ **do**
8:         **if** $localNo \notin map$ **then**
9:             $i \leftarrow \max(globCls) + 1$
10:             $map[localNo] \leftarrow i$
11:         **end if**
12:         $globCls[mPos] \leftarrow map[localNo]$
13:     **end for**
14: **end for**

---

of each mention in these lists. The overlapping structure of consecutive inputs (Figure 4), enables the construction of a mapping dictionary (i.e., `map`), between the local and global cluster numbers across consecutive frames (i.e., second frame of the kth input and first frame of the (k+1)[th] input). For each input (e.g., frame tuples), global cluster numbers that are established for the first frame (`before`) are stored into the mapping dictionary. Then, for the second frame, if any local cluster number has not yet been assigned to a global cluster number, then a new cluster is created with number i, and the mention is inserted to this newly created cluster. At the end, mentions from the second frame, are inserted into a global cluster.

3.3 Controlled Inference Method

After an LLM is fine-tuned using the aforementioned tuples (i.e., <instruction, masked input, output>), it learns and adapts to the coreference linking process. Specifically, the model becomes capable of linking overt and zero mentions with the assistance of the developed data processors.

The coreferential relations of each mention are resolved sequentially as they appear in an input sentence during both training and inference. Due to the causal nature of LLMs, these models correlate information in an autoregressive manner for each decision, starting from the current mention and moving towards the descendant mentions and their predicted cluster numbers. Consequently, we developed a controlled inference methodology to mitigate the model's tendency to make random

```
                    [INSTRUCTION]
-----------------------------------------------------------------
                      [INPUT]
<m> Jake </m>#MASK goes to school. <m> Amy </m>#MASK and <m>
Jake </m>#MASK ...
-----------------------------------------------------------------
                      [OUTPUT]

<m> Jake </m>#↑

<m> Jake </m>#0 goes to school. <m> Amy </m>#↑

<m> Jake </m>#0 goes to school. <m> Amy </m>#1 and <m> Jake </m>#↑

<m> Jake </m>#0 goes to school. <m> Amy </m>#1 and <m> Jake </m>#0
```
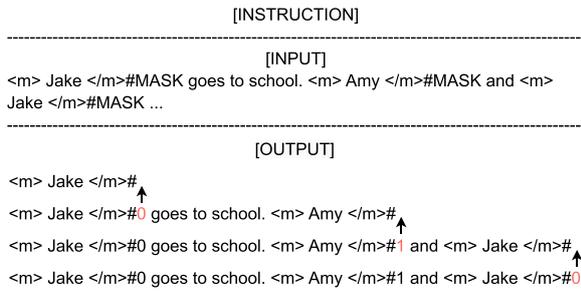
Figure 5: Inference steps.

predictions when formulating new decisions during the inference phase, and to ensure that it relies on prior context for its decision-making. This approach focuses solely on predicting the cluster number to replace the MASK token in the input, while automatically keeping all other tokens the same in the original sentence.

The inference steps for English samples are illustrated in the [OUTPUT] section of Figure 5, which also includes the masked input sentences under the [INPUT] section. Each inference step is presented on a new line in the [OUTPUT] section. The arrow shows the position of MASK tokens where the model is called for inference. The model predicts the current MASK token by considering all preceding context up to the start of the output. The non-colored parts, namely up to MASK, are given to the model and then the red part is generated by the model during the inference. For example, for the first MASK token, the model generates a cluster number 0, immediately after the position indicated by the arrow, by replacing the current MASK token with the predicted cluster number. In each inference step, the pre-context is updated with a newly predicted cluster number to maintain the flow of cluster information in subsequent inference steps. The arrow then moves to the next MASK token, reaching the position where the model will be called again. In this way, all MASK tokens are predicted sequentially, allowing the model to utilize its previous decisions for the current decision.

This inference mechanism is termed the controlled inference strategy which provides several advantages: 1) It significantly reduces computational load during inference by calling the model on only at specified points, leading to faster processing times, 2) It maintains the original text structure, ensuring consistency in input/output formats and enabling easy conversion to the desired

data formats. By focusing the model's efforts exclusively on predicting cluster number, the proposed framework achieves a balance between accuracy, speed, and practical applicability.

4 Experimental Setup & Results

This section introduces the details of the experimental setup for the fine-tuning process, as well as the performances of the tested models.

4.1 Experimental Setup

This study validates the proposed methodology on instruction-tuned versions of three recent LLMs: Llama 3.1,[5] Gemma 2,[6] and Mistral 0.3.[7] These models are fine-tuned using instruction-based method specifically for the coreference linking phase in CR. Low-Rank Adaptation (LoRA) (Hu et al., 2021) with a rank of 16 (R = 16), and 4-byte quantization is used during training (i.e., few-shot and full) to use resources efficiently. This setup is designed to balance computational efficiency with model performance, leveraging the capabilities of the LLM while maintaining tractable training times and resource requirements. In this study, full training denotes training the LLM on all samples within a dataset, whereas few-shot (specifically 5-shot) training involves training using only 5 examples from each dataset. For the initial experiments, training and validation splits are chosen from a subset of an English dataset (en_gum) from CorefUD v1.2. The LLMs are then fully trained with 3 different seeds (i.e., 3 models for each LLM-instruction set pair in total) and subsequently evaluated for each instruction set using these data splits. The average performance across these runs is presented as the final performance for each LLM-instruction set pair. This approach is essential to enhance the reliability of our preliminary results. The full training procedure is then conducted on the entire CorefUD v1.2 using the highest-performing LLM-instruction set pair, which is accepted as the proposed model in this study, CorefInst. All models are trained and evaluated on a single NVIDIA V100 (32 GB) or A100 (80 GB) GPU.

---

[5]https://huggingface.co/unsloth/Meta-Llama-3.1-8B-Instruct-bnb-4bit.

[6]https://huggingface.co/unsloth/gemma-2-9b-it-bnb-4bit.

[7]https://huggingface.co/unsloth/mistral-7b-instruct-v0.3-bnb-4bit.

The CorefUD v1.2 dataset collection (Popel et al., 2024) is utilized for the training and evaluation of the models under investigation. This dataset is multilingual and comprises documents formatted in CoNLL-U, along with coreference annotations. The CorefUD v1.2 contains 21 datasets from 15 languages, also containing pro-dropped languages with zero-mention annotations (see Table 5 for the dataset statistics).

The initial experiments on a subset of the English documents take ∼20 minutes on a V100 GPU (batch size: 4, epochs: 5). The 5-shot training takes 10 minutes (batch size: 16, epochs: 5), while full training on CorefUD v1.2 takes ∼15 hours on an A100 GPU (batch size: 16, epochs: 3). The pre-defined length of each frame is up to 1600 and the total of an input tuple is up to 7168 for the proposed model.

CorPipe24 (Straka, 2024), the most advanced version of the CorPipe system family and the winner of the latest multilingual CR shared task (Novák et al., 2024), is used as the leading multilingual CR model for comparing our proposed model in an end-to-end manner. We used its single-stage variant, which employs a single pre-trained encoder model to jointly predict empty nodes (i.e., zero mentions), coreferential overt mentions, and coreferential links between them. This variant is the large-sized, single-stage model checkpoint that achieves the highest development performance across all CorefUD v1.2 datasets. Since the proposed model in this study was also selected using the same approach, comparing it with the single-variant of the CorPipe24[8] system provides a fairer comparison. For end-to-end evaluation of CorefInst (i.e., on predicted overt mentions and zeros), the mention/zero detection output from CorPipe24 is directly used, ensuring both models operate on the same predicted input during the coreference linking stage. Similar to the initial experiments, during the end-to-end evaluation of our proposed model, CorefInst, and its comparison with CorPipe24, all models were trained and evaluated using 3 different random seeds. The final performances reported for each model are presented as the average of these 3 runs. Moreover, we apply statistical significance testing (SST) to the performance differences between

	Llama 3.1	Gemma 2	Mistral 0.3
Inst. #1	$75.57_{\pm0.49}$	$72.45_{\pm1.71}$	$\mathbf{72.46}_{\pm\mathbf{1.37}}$
Inst. #2	$74.71_{\pm0.32}$	$73.51_{\pm0.84}$	$71.28_{\pm1.64}$
Inst. #3	$74.91_{\pm1.35}$	$\mathbf{73.82}_{\pm\mathbf{2.13}}$	$71.44_{\pm2.59}$
Inst. #4	$75.54_{\pm1.48}$	$73.43_{\pm0.96}$	$71.58_{\pm2.15}$
Inst. #5	$\mathbf{76.15}_{\pm\mathbf{1.16}}$	$73.25_{\pm1.07}$	$71.55_{\pm2.17}$
AVG	$\mathbf{75.38}_{\pm\mathbf{0.57}}$	$73.29_{\pm0.52}$	$71.66_{\pm0.46}$

Table 1: The performances of the initial experiments with a subset of English dataset.

these models[9] to show the robustness of the performance both in terms of the average and across the datasets in each language. We apply the Almost Stochastic Order (ASO) test with confidence level $\alpha = 0.95$ (Ulmer et al., 2022) on multiple runs (i.e., 3 runs) with random seeds.

The best model is evaluated on the development splits of the CorefUD v1.2 since test splits are blind. The average CoNLL coreference metric which is the average of three different metrics: MUC, B-Cubed, and entity-based CEAF are used as the main performance metric. One should note that, these CR metrics (MUC, B^3, $CEAF_e$, and CoNLL—the average of the three) evaluate the quality of mention clustering (i.e., entitites) from different perspectives, categorized as link-based, mention-based, and entity-based metrics, respectively. None of these metrics were specifically designed to evaluate the coreference performance of dropped pronouns. However, with the CorefUD initiative, CR studies on pro-drop languages have gained greater significance. In this context, the anaphor-decomposable score (i.e., zero metric) has been recently proposed (Žabokrtský et al., 2022) as a dedicated metric for evaluating a model on coreferences arising from zero mentions. As a final evaluation, the performance of the best model is also presented with this metric. An available CR scorer[10] is used for the evaluation to be sure that the performances are compatible with the literature.

4.2 Results & Discussion

The results of the initial experiments on a subset of the English dataset is provided in Table 1, where

---

	System	AVG	ca_ancora	cs_pcedt	cs_pdt	cu_proiel	de_parcor	de_potsdam	en_gum	en_litbank	en_parcor	es_ancora	fr_democrat	grc_proiel	hbo_ptnk	hu_korkor	hu_szeged	lt_lcc	no_bokmaal	no_nynorsk	pl_pcc	ru_rcr	tr_itcc
GOLD	CorefInst$_{few}$	73.8	77.1	74.0	70.3	56.7	81.8	76.3	69.7	76.7	83.4	76.4	67.3	72.2	65.5	77.2	82.7	93.3	67.2	64.1	67.4	84.7	66.0
GOLD	CorefInst$_{full}$	**85.8**	**92.0**	**90.4**	**87.6**	**76.1**	**91.1**	**85.5**	**81.8**	**84.5**	**95.2**	**93.0**	**77.1**	**85.7**	**71.1**	**86.7**	**91.7**	**96.7**	**81.3**	**82.2**	**82.3**	**90.0**	**80.5**
PRED	CorPipe24	70.5	78.9	71.0	74.3	50.3	**70.8**	75.4	74.4	74.5	66.2	79.7	**69.6**	62.8	**65.5**	62.2	65.9	**79.5**	**80.3**	**80.1**	73.4	75.4	50.1
PRED	CorefInst$_{full}$	**72.3**	**79.5**	**71.3**	**75.8**	**57.0**	68.4	**77.9**	**75.5**	**78.8**	**68.0**	**80.8**	69.4	**70.7**	63.9	**64.1**	**67.1**	79.1	78.9	78.4	**75.9**	**75.9**	**61.6**

Table 2: The performance of the CorefInst on gold and predicted mentions for each individual language.

System	AVG	ca_ancora	cs_pcedt	cs_pdt	cu_proiel	es_ancora	grc_proiel	hu_korkor	hu_szeged	pl_pcc	tr_itcc
CorPipe24	68.7	77.7	56.6	70.2	60.6	79.3	75.1	67.1	63.9	77.9	58.2
CorefInst$_{full}$	**77.1**	**86.3**	**61.5**	**76.1**	**71.8**	**88.0**	**85.8**	**70.4**	**68.7**	**85.0**	**77.7**

*These scores are not directly comparable with the CoNLL scores on entire entities in Table 2.

Table 3: Anaphor-decomposable F1 scores* on predicted zero anaphors.

each inner cell shows the average performance of 3 different runs (with also standard deviation) for each LLM-instruction set pair with different seeds, and the last row reports the average performances of the LLMs across instruction sets, along with the standard deviation. The best performance of Llama is obtained by Inst. #5 (76.15% CoNLL), whereas Gemma and Mistral reach their highest performance with Inst. #3 and Inst. #1 (73.82% and 72.46%, respectively). When the overall table is examined, the lowest average performance is obtained with Mistral, followed by Gemma. Llama achieves the highest performance for all instructions as well as on average across instructions. Based on these initial results, it has been decided to continue with the highest scoring LLM-instruction pair (Llama-Inst. #5) for further analysis.

The top-performing model (hereafter, CorefInst in Tables 2 and 3) is then 5-shot or fully trained and evaluated on CorefUD v1.2. Table 2 shows the performances of the CorefInst across various configurations. Firstly, the model's performance is analyzed following fine-tuning using both 5-shot (CorefInst$_{few}$) and full training (CorefInst$_{full}$) strategies on gold mentions. As seen, the CorefInst$_{full}$ outperforms the CorefInst$_{few}$ in each language and their all documents, and also in average by 12 percentage points (pp) across all languages. The improvements across all datasets range from 3.4 to 19.4, as obtained on Lithunian

lt_lcc) and Antic Slavonic (cu_proiel). The top five languages where the model performed the best are Lithuanian, Russian (ru_rcr), Ancient Hebrew (hbo_ptnk), English (en_litbank), and Hungarian (hu_szeged). In addition, in the perspective of pro-dropped languages, which are Catalan (ca_ancora), Czech (cs_[pcedt,pdt]), Slavonic, Spanish (es_ancora), Greek (grc_proiel), Hungarian (hu_[korkor,szeged]), Polish (pl_pcc), and Turkish (tr_itcc), CorefInst$_{full}$ also outperforms its few-shot counterparts.

The performance of CorefInst$_{full}$ on the predicted mentions is also analyzed, enabling the end-to-end evaluation of the proposed model, as indicated in the last row of Table 2. The third line of the table also includes the performance of CorPipe24 for comparison. As a reminder, as part of the end-to-end evaluation, the models' performances are reported by the average results across three runs of models. CorefInst$_{full}$ achieves 1.8 pp improvement than CorPipe24 on predicted overt and zero mentions in average (70.5$_{\pm1.1}$% vs. 72.3$_{\pm1.3}$%), which is statistically significant. CorefInst$_{full}$ outperforms CorPipe24 on several languages, with the exception of one dataset in German (de_parcor) and as well as datasets in French, Ancient Hebrew, Lithuanian and Norwegian. Among improvements obtained by CorefInst$_{full}$, the increases observed on the Catalan, Czech (cs_pdt), Old Church Slavonic, German (de_potsdam),

English (en_gum, en_litbank), Spanish, Ancient Greek (grc_proiel), Hungarian (hu_szeged), Polish (pl_pcc), and Turkish are statistically significant, whereas the improvements on one dataset each in Czech (cs_pcedt), English (en_parcor), and Hungarian (hu_korkor), as well as on Russian, are not statistically significant. The 5-top significant highest increments are observed on Turkish (11.5 pp), Ancient Greek (7.9 pp), Old Church Slavonic (6.7 pp), English (4.8 pp, en_litbank), and German (de_potsdam) and Polish (2.5 pp for both).

As a final exploration, in Table 3, we also report the average performance of three runs of the proposed model (CorefInst$_{full}$) on predicted zero mentions, evaluated across the related datasets of pro-drop languages. This evaluation indicates how well the model does from the perspective of zero mentions. In addition to achieving higher performance in the end-to-end evaluation using the CoNLL metric on predicted overt and zero mentions (Table 2), the proposed model also demonstrates higher coreference linking performance on predicted zeros across all pro-dropped languages, improving the average performance by 8.4 pp ($68.7_{\pm0.9}\%$ vs. $77.1_{\pm1.4}\%$), which is statistically significant. On a language-specific basis, all improvements are statistically significant, ranging from 3.3 pp (for Hungarian, hu_korkor) to 19.5 pp (for Turkish). Zero mentions are implicit references to an entity within a sentence, so that resolving their coreferential relations necessitates a comprehensive semantic understanding of the entire text. This experiment demonstrates that LLMs, pre-trained on diverse and extensive datasets, exhibit greater capability in resolving hidden semantic relations in a text arising from zero mentions, compared to CR-specific neural architectures.

5   Conclusion

This article presents the first approach for instruction-based fine-tuning of an LLM for the multilingual coreference resolution (CR) task, uniquely incorporating both overt and zero mentions (i.e., coreferential dropped pronouns). The proposed framework includes comprehensive data processing and mention clustering stages, and a controlled inference method designed to meet the strict data constraints and relation resolution phase.

The proposed framework is evaluated across three LLMs, Llama 3.1, Gemma 2, and Mistral 0.3, with five distinct instruction sets. In-depth analysis of the best-performing LLM-instruction set pair (i.e., Llama 3.1 with the fifth instruction) is performed under the settings of few-shot and full training on gold mentions, along with an end-to-end evaluation on predicted mentions including both overt and zero ones. The findings suggest that LLMs, when fine-tuned with an appropriate instruction set, can achieve superior performance compared to state-of-the-art task-specific architectures. Results reveal that the best model (i.e., fully trained Llama 3.1-Inst. #5, CorefInst$_{full}$) achieves an 12 pp gain in average on gold mentions across 21 documents in 15 languages over its few-shot version. Furthermore, this model outperforms CorPipe24, the state-of-the-art end-to-end multilingual coreference resolution model on CorefUD v1.2, by an average of 1.8 percentage points which is statistical significant. Moreover, statistically significant language-specific improvements range from 0.6 to 11.5 percentage points. This study also demonstrates that decoder-only models, when properly fine-tuned, can effectively resolve zero anaphors which requires a deep understanding of the contextual focus. These findings show the effectiveness of instruction-tuning for flexible, scalable, and high-performing multilingual CR. At the end, this study provides practical insights to guide future research focused on scaling multilingual CR through general-purpose LLMs instead of CR-specific architectures.

Limitations

The main limitation of this study is that coreferential relations throughout the document rely on inter-frame resolved coreferential relations within an input tuple. For an entity cluster, the chain of coreferential relations is established via overlapping frames of consecutive input tuples. This may cause some relations to not be carried further in the document due to a false-negative relation between frames within an input tuple. In addition, the absence of any mentions in these overlapping frames may also disrupt the coreferential relation chain, leading to the splitting of an entity's mention cluster into sub-clusters, each containing mentions referring to the same entity.

Due to the size and the time and resource complexity required for fine-tuning of the LLMs explored, they could only be fine-tuned with LoRA for the proposed methodology. Therefore, the starting points for the LLMs explored are already their quantized versions.

Studies using LLMs acknowledge the potential limitation arising from the possibility that pretrained LLMs may have been exposed to texts in the evaluation datasets of the tasks in focus during their pretraining phase. This situation might be possible in our study as well; however, the novelty of this paper lies in modeling how this problem can be approached for the first time in the literature. Therefore, even if the LLMs were exposed to the evaluation dataset's text, it is unlikely that they have seen it in a way that would be used for instruction-based fine-tuning. Although we didn't provide results in a zero-shot setting, our 5-shot results, which are significantly behind the fully trained model, provide enough clues that pretrained LLMs are not yet capable of solving this complicated CR task.

This study proposed a framework for only coreference linking stage for the CR. The main reason for this is that the mention detection framework planned to be proposed allows the start and end tags for each mention to be placed at any position in the sentence, which requires a long inference time. Developing an effective, accurate, and efficient controlled inference mechanism for this stage is included as future work in this study.

During the revision period of this article, new open-source decoder-only LLMs, such as Bi et al. (2024), have been released, reflecting the rapid development in the field. Due to the computational cost of extensive evaluations, we prioritized maintaining the scope of our experiments while acknowledging the potential for future work on newer models. Since the proposed model is not constrained by a task-specific architecture, future studies may extend our findings by integrating and evaluating this approach with newly developed decoder-only LLMs to further investigate their effectiveness in CR.

Due to the high computational cost of training and evaluating all possible model-instruction combinations on the full dataset, only the top-performing pair from preliminary experiments was selected for further evaluation. Although multiple runs were conducted to ensure the reliability of initial results, we acknowledge that differences among instructions were relatively small, and it is possible that other instructions may lead to better performance. Exploring different instructions more thoroughly across various languages and LLMs could be a promising area for future work.

Other suggested research directions for multilingual CR include investigating the impact of larger context windows on performance, conducting controlled architectural comparisons between decoder-only and encoder-based LLMs, and examining the tradeoffs between English-centric and multilingual models. These studies could provide deeper insights and contribute to advancing multilingual CR beyond the scope of the current work.

Acknowledgments

References

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34. https://doi.org/10.1162/coli.2008.34.1.1

Santanu Bhattacharjee, Rejwanul Haque, Gideon Maillette de Buy Wenniger, and Andy Way. 2020. Investigating query expansion and coreference resolution in question answering on BERT. In *International Conference on Applications of Natural Language to Information Systems*, pages 47–59. Springer.

`https://doi.org/10.1007/978-3-030-51310-8_5`

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, A. X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. 2024. DeepSeek LLM: Scaling open-source language models with longtermism. arxiv 2024. *arXiv preprint arXiv:2401.02954*.

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. `https://doi.org/10.1162/tacl_a_00543`

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clar, K. Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Shisong Chen, Binbin Gu, Jianfeng Qu, Zhixu Li, An Liu, Lei Zhao, and Zhigang Chen. 2021. Tackling zero pronoun resolution and non-zero coreference resolution jointly. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 518–527, Online. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2021.conll-1.40`

Kevin Clark and Christopher D. Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Berlin, Germany. Association for Computational Linguistics. `https://doi.org/10.18653/v1/P16-1061`

Marie-Catherine De Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Computational Linguistics*, 47(2):255–308. `https://doi.org/10.1162/coli_a_00402`

Karel D'Oosterlinck, Semere Kiros Bitew, Brandon Papineau, Christopher Potts, Thomas Demeester, and Chris Develder. 2023. CAWcoref: Conjunction-aware word-level coreference resolution. In *Proceedings of The Sixth Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2023)*, pages 8–14. `https://doi.org/10.18653/v1/2023.crac-main.2`

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.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models.

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. `https://doi.org/10.1162/tacl_a_00300`

Ben Kantor and Amir Globerson. 2019. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677. `https://doi.org/10.18653/v1/P19-1066`

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. Association for Computational Linguistics. `https://doi.org/10.18653/v1/D17-1018`

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*, pages 687–692. Association for Computational Linguistics.

Houjun Liu, John Bauer, Karel D'Oosterlinck, Christopher Potts, and Christopher D. Manning. 2024. MSCAW-coref: Multilingual, singleton and conjunction-aware word-level coreference resolution. In *Proceedings of The Seventh Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 33–40. `https://doi.org/10.18653/v1/2024.crac-1.4`

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. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2022.findings-emnlp.70`

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. `https://doi.org/10.18653/v1/2021.sigdial-1.53`

Anna Nedoluzhko, Michal Novák, Martin Popel, Zdenek Žabokrtský, Amir Zeldes, and Daniel Zeman. 2022. CorefUD 1.0: Coreference meets universal dependencies. In *Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022)*, pages 4859–4872. European Language Resources Association.

Michal Novák, Barbora Dohnalová, Miloslav Konopik, Anna Nedoluzhko, Martin Popel, Ondrej Prazak, Jakub Sido, Milan Straka, Zdeněk Žabokrtský, and Daniel Zeman. 2024. Findings of the third shared task on multilingual coreference resolution. In *Proceedings of The Seventh Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 78–96. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2024.crac-1.8`

Tuğba Pamay Arslan, Kutay Acar, and Gülşen Eryiğit. 2023. Neural end-to-end coreference resolution using morphological information. In *Proceedings of the CRAC 2023 Shared Task on Multilingual Coreference Resolution*, pages 34–40. `https://doi.org/10.18653/v1/2023.crac-sharedtask.3`

Tugba Pamay Arslan and Gülsen Eryiğit. 2025. Enhancing Turkish coreference resolution: Insights from deep learning, dropped pronouns, and multilingual transfer learning. *Computer Speech & Language*, 89:101681. `https://doi.org/10.1016/j.csl.2024.101681`

Martin Popel, Michal Novák, Zdeněk Žabokrtský, Daniel Zeman, Anna Nedoluzhko, Kutay Acar, David Bamman, Peter Bourgonje, Silvie Cinková, Hanne Eckhoff, Gülşen Cebiroğlu Eryiğit, Jan Hajič, Christian Hardmeier, Dag Haug, Tollef Jørgensen, Andre Kåsen, Pauline Krielke, Frédéric Landragin, Ekaterina Lapshinova-Koltunski, Petter Mæhlum, M. Antònia Martí, Marie Mikulová, Anders Nøklestad, Maciej Ogrodniczuk, Lilja Øvrelid, Tuğba Pamay Arslan, Marta Recasens, Per Erik Solberg, Manfred Stede, Milan Straka, Daniel Swanson, Svetlana Toldova, Noémi Vadász, Erik Velldal, Veronika Vincze, Amir Zeldes, and Voldemaras Žitkus. 2024. Coreference in

universal dependencies 1.2 (CorefUD 1.2). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint conference on EMNLP and CoNLL-shared Task*, pages 1–40.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Natalia Skachkova. 2024. Multilingual coreference resolution as text generation. In *Proceedings of The Seventh Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 114–122. https://doi.org/10.18653/v1/2024.crac-1.11

Natalia Skachkova, Tatiana Anikina, and Anna Mokhova. 2023. Multilingual coreference resolution: Adapt and generate. In *Proceedings of the CRAC 2023 Shared Task on Multilingual Coreference Resolution*, pages 19–33. https://doi.org/10.18653/v1/2023.crac-sharedtask.2

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. https://doi.org/10.18653/v1/W18-6306

Milan Straka. 2023. ÚFAL CorPipe at CRAC 2023: Larger context improves multilingual coreference resolution. In *Proceedings of the CRAC 2023 Shared Task on Multilingual Coreference Resolution*, pages 41–51, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.crac-sharedtask.4

Milan Straka. 2024. CorPipe at CRAC 2024: Predicting zero mentions from raw text. In *Proceedings of the Seventh Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 97–106, Miami. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.crac-1.9

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LlaMa: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Dennis Ulmer, Christian Hardmeier, and Jes Frellsen. 2022. Deep-significance-easy and meaningful statistical significance testing in the age of neural networks. *arXiv preprint arXiv:2204.06815*.

Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. 2016. Learning global features for coreference resolution. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 994–1004. Association for Computational Linguistics. https://doi.org/10.18653/v1/N16-1114

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. `https://doi.org/10.18653/v1/2020.acl-main.622`

Xiaohan Yang, Eduardo Peynetti, Vasco Meerman, and Chris Tanner. 2022. What GPT knows about who is who. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 75–81, Dublin, Ireland. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2022.insights-1.10`

Gokul Yenduri, M. Ramalingam, G. Chemmalar Selvi, Y. Supriya, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, G. Deepti Raj, Rutvij H. Jhaveri, B. Prabadevi, Weizheng Wang, Athanasios V. Vasilakos, and Thippa Reddy Gadekallu. 2024. Generative Pre-trained Transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE Access*. `https://doi.org/10.1109/ACCESS.2024.3389497`

Zdeněk Žabokrtský, Miloslav Konopík, Anna Nedoluzhko, Michal Novák, Maciej Ogrodniczuk, Martin Popel, Ondřej Pražák, Jakub Sido, Daniel Zeman, and Yilun Zhu. 2022. Findings of the shared task on multilingual coreference resolution. In *Proceedings of the CRAC 2022 Shared Task on Multilingual Coreference Resolution*, pages 1–17. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2023.crac-sharedtask.1`

Zdeněk Žabokrtský, Miloslav Konopik, Anna Nedoluzhko, Michal Novák, Maciej Ogrodniczuk, Martin Popel, Ondrej Prazak, Jakub Sido, and Daniel Zeman. 2023. Findings of the second shared task on multilingual coreference resolution. In *Proceedings of the CRAC 2023 Shared Task on Multilingual Coreference Resolution*, pages 1–18. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2023.crac-sharedtask.1`

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. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2023.emnlp-main.704`

Appendix

Inst. #1	The text is in English.
	You are a coreference resolver.
	Rewrite the sentence considering these rules:
	Mentions are in <m>...</m>#MASK format.
	Group the mentions that refer to same real-world entity.
	If mentions refer to same thing write the same number instead of MASK.
	If mentions represent different things write another number.
	Where you see </z>@ there is a zero mention, which is normally not written but you also need to link them with other mentions.
Inst. #2	Identify instances of coreference where different mentions refer to the same entity.
	Rewrite the passage, tagging each term with a unique identifier and indicating coreferential relationships.
	Ensure accuracy and consider context.
	Fill MASK with unique number for every entity. Be sure that they represent exactly the same entity, not similar.
	Example output format:
	. . . <m>Bertrand Russell </m>#MASK is a good author, I love <m>The History of Western Philosophy </m>#MASK ...
	. . . <m>Bertrand Russell </m>#0 is good author, I love <m>The History of Western Philosophy </m>#1 . . .
	Where you see </z>@ there is a zero mention, which is normally not written but you also need to link them with other mentions.
Inst. #3	For every mention in <m></m>tags examine if there is any coreferential/coherent mention.
	If two mentions represent the same entity write the same number instead of MASK after closing mention tag </m>.
	Do not change anything else other than MASK.
	Where you see </z>@ there is a zero mention, which is normally not written but you also need to link them with other mentions.
Inst. #4	For every mention in <m></m>tags examine if there is any coreferential mention.
	If two mentions represent the same entity write the same number instead of MASK after closing mention tag </m>.
	For example: author and book represent different entities.
	Do not change anything else other than MASK.
	Where you see </z>@ there is a zero mention, which is normally not written but you also need to link them with other mentions.
Inst. #5	For every mention in <m></m>tags examine if there is any coreferential/coherent mention.
	If two mentions represent the same entity write the same number instead of MASK after closing mention tag </m>.
	For example: author and book represent different entities.
	Do not change anything else other than MASK.
	Where you see </z>@ there is a zero mention, which is normally not written but you also need to link them with other mentions.

Table 4: The list of instruction sets taken into consideration.

Document	Total number of					Distribution	
	docs	sents	words	empty	mentions	train %	dev %
ca_ancora	1142	12K	387K	6K	56K	87.18	12.82
cs_pcedt	2212	47K	1,13M	34K	163K	85.08	14.92
cs_pdt	2849	44K	761K	19K	166K	88.10	11.90
cu_proiel	19	6K	61K	6K	20K	86.01	13.99
de_parcorfull	17	1K	910K	0	1K	88.73	11.27
de_potsdamcc	159	2K	30K	0	5K	88.77	11.23
en_gum	191	11K	188K	100	53K	87.28	12.72
en_litbank	90	8K	189K	0	26K	88.78	11.22
en_parcorfull	17	1K	10K	0	1K	88.36	11.64
es_ancora	1211	13K	420K	7K	64K	88.89	11.11
fr_democrat	96	11K	256K	0	70K	88.97	11.03
grc_proiel	14	6K	65K	6K	20K	93.95	6.05
hbo_ptnk	28	1K	18K	0	5K	42.32	57.68
hu_korkor	85	1K	24K	2K	4K	88.58	11.42
hu_szegedkoref	360	8K	117K	4K	14K	89.42	10.58
lt_lcc	90	2K	33K	0	4K	89.89	10.11
no_bokmaalnarc	315	14K	225K	0	68K	90.37	9.63
no_nynorsknarc	364	11K	191K	0	57K	90.58	9.42
pl_pcc	1646	32K	502K	17K	169K	88.89	11.11
ru_rucor	163	8K	145K	0	15K	85.39	14.61
tr_itcc	21	4K	60K	10K	19K	90.36	9.64

Table 5: CorefUD v1.2 statistics (Novák et al., 2024). The statistics for sentences, words, empty, and mentions are replaced with approximate counts by rounding up the numbers.