

## EXPLORING MULTIPLE STRATEGIES TO IMPROVE MULTILINGUAL COREFERENCE RESOLUTION IN COREFUD

Ondřej PRAŽÁK, Miloslav KONOPÍK, Pavel KRÁL

*NTIS – New Technologies for the Information Society*  
*Faculty of Applied Sciences, University of West Bohemia*  
*Technická 8, Plzeň, Czech Republic*  
*e-mail: ondafa@ntis.zcu.cz, {konopik, pkral}@kiv.zcu.cz*

**Abstract.** Coreference resolution, the task of identifying expressions in a text that refer to the same entity, is a critical component in various natural language processing applications. This paper presents a novel end-to-end neural coreference resolution system, utilizing the CoreFUD 1.1 dataset, which spans 17 datasets across 12 languages. The proposed model is based on the standard end-to-end neural coreference resolution system. We first establish baseline models, including monolingual and cross-lingual variations, and then propose several extensions to enhance performance across diverse linguistic contexts. These extensions include cross-lingual training, incorporation of syntactic information, a Span2Head model for optimized headword prediction, and advanced singleton modeling. We also experiment with headword span representation and long-document modeling through overlapping segments. The proposed extensions, particularly the heads-only approach, singleton modeling, and long document prediction, significantly improve performance across most datasets. We also perform zero-shot cross-lingual experiments, highlighting the potential and limitations of cross-lingual transfer in coreference resolution. Our findings contribute to the development of robust and scalable coreference systems for multilingual coreference resolution. Finally, we evaluate our model on the CoreFUD 1.1 test set and surpass the best model from the CRAC 2023 shared task of comparable size by a large margin.

**Keywords:** Coreference resolution, cross-lingual model, transformers, end-to-end model

**Mathematics Subject Classification 2010:** 68T50

## 1 INTRODUCTION

Coreference resolution is the task of identifying which expressions in a text refer to the same entity in the real world. The task can be divided into two subtasks. Identify entity mentions and group them according to the entities they refer to. Anaphora resolution is closely related to coreference resolution. Anaphora is a contextual reference to an expression in the previous context. Most of the anaphoric relations are also coreferential, but it does not necessarily have to be so. For example, in the sentence:

**This apple** is mine. Please, take another **one**.

The two mentions are anaphoric, but they are not coreferential (they refer to different apples). Some datasets contain only anaphora, some only coreferences, and others contain annotations of both. Anaphoric relations are based more on the syntax of a sentence, whereas coreference relations are more about semantics. Coreferences are usually annotated in terms of coreference clusters or in terms of coreference chains. Coreference clusters group together all the mentions of the same entity, no matter their order. In the coreference chain, each mention is aligned with its direct antecedent (closest previous mention of the same entity). Although significant advancements have been made in coreference resolution for English, developing robust and effective models for multiple languages remains a challenging task due to linguistic diversity and the availability of resources.

Historically, coreference resolution was a standard preprocessing step in various natural language processing (NLP) tasks, such as machine translation, summarization, and information extraction. Although recent large language models have achieved state-of-the-art results in coreference resolution, they are expensive to train and deploy, and traditional (discriminative) approaches remain competitive. Expressing this task in natural language is challenging, and to the best of our knowledge, there have been no successful attempts to utilize large chatbots (like ChatGPT-4) to achieve superior results.

Coreference resolution becomes particularly challenging in low-resource languages. One strategy to address this challenge is to train a multilingual model on datasets from multiple languages, thereby transferring knowledge from resource-rich languages to those with fewer resources. However, a significant challenge with this approach lies in the differences in annotations across available corpora. The CorefUD initiative [1] tries to harmonize the datasets and create one annotation scheme for coreference in multiple languages similarly to Universal Dependencies [2] for syntactic annotations.

This paper proposes a novel approach to multilingual coreference resolution. We use the CorefUD 1.1 dataset [1]. The CorefUD 1.1 is a harmonized corpus with 17 different datasets spanning 12 languages. The dataset is intended to extend Universal Dependencies [2] with coreference annotations. Therefore, all datasets within CorefUD are treebanks. Coreference annotations are built upon these dependen-

cies, meaning that mentions are represented as subtrees in the dependency tree and can be captured by their heads. Moreover, the evaluation metric used with this dataset on the CRAC 2023 Multilingual Coreference Resolution Shared Task uses head-match evaluation. This means that the predicted mention is considered the same as the gold mention if they share the same head. Notable differences exist between the datasets, with one of the most prominent being the presence of singletons. Singletons are clusters that contain only one mention and do not participate in any coreference relation, yet they are still annotated as mentions. For further details on the dataset, refer to [1].

Building upon the baseline model proposed by [3], we introduce several novel extensions aimed at enhancing the performance of coreference resolution across multiple languages and datasets within the CorefUD collection. Our primary goal is to develop a universal model capable of handling the diverse and complex nature of these datasets effectively.

First, recognizing the challenge posed by small dataset sizes in the CorefUD collection, we propose a cross-lingual training approach. By pretraining our model on a concatenated dataset that includes all available training data across languages, we aim to improve the model’s ability to generalize across languages. This approach is particularly beneficial for low-resource languages, where training large models from scratch is impractical due to the limited data available.

Next, we incorporate syntactic information into our model to leverage the dependency structures inherent in the CorefUD datasets. This extension adds syntax to the token representations by encoding their paths to the ROOT in the dependency tree, thereby enriching the model’s understanding of syntactic relationships, which are crucial for accurate mention detection and coreference resolution.

In response to the evaluation metrics used in the CRAC 2022 Shared Task, we also evaluate the Span2head model, which simplifies span-based mention representation to headword prediction. This adjustment aligns the model’s output more closely with the evaluation criteria, where we consider two mentions the same if they share the same head.

Additionally, we explore the potential of head representations as a simplified approach to coreference resolution. Similarly to word-level coreference [4], by modeling mentions solely based on their syntactic heads, we reduce the computational complexity from quadratic to linear, making the model more efficient and less prone to errors, particularly in the case of long and complex mentions.

We further address the issue of singleton – mentions by introducing mechanisms to incorporate these into the training process. By modeling singletons explicitly, we ensure that valuable training data from singleton-rich datasets is not discarded, thus improving the model’s robustness and accuracy across different datasets.

Finally, to overcome the limitations of short sequence lengths in XLM-R, we propose an approach that utilizes overlapping segments with a cluster merging algorithm. This method ensures that coreference chains spanning multiple segments

are correctly identified and merged, even when document segmentation is necessary due to memory constraints.

We experimentally evaluate all proposed extensions to find the best configuration for a joint model for all the datasets in CorefUD. Through these extensions, our model aims to advance the state of multilingual coreference resolution by addressing the specific challenges posed by the diverse datasets in the CorefUD collection, ultimately contributing to more accurate and generalizable coreference systems.

The main contributions of the paper are as follows:

- We propose several extensions of a standard model that lead to significant improvements. The most prominent extensions include singletons modeling, head-word mention representation, and segments overlapping for long coreference prediction.
- We provide a detailed evaluation of all proposed extensions.
- We perform the analysis of knowledge transferability between the datasets with joint training and cross-lingual zero-shot experiments.
- We demonstrated that the proposed model significantly surpasses the state of the art with a comparable model size.
- We trained a single model that has a great performance on all the datasets in the CorefUD collection.
- Our source codes are publicly available on GitHub for subsequent research: <https://github.com/ondfa/coref-multiling>.

## 2 RELATED WORK

### 2.1 End-to-End Neural Coreference Resolution Models

The first end-to-end neural coreference resolution system was introduced by [5], and many subsequent neural coreference resolution systems are based on their model.

The model estimates the probability  $P(y_i|D)$  of a mention  $i$  corefering with the antecedent  $y_i$  in a document  $D$ . The model is trained in an end-to-end manner, so every continuous span up to the maximum length is considered as a mention. Therefore, we work with  $N = \frac{T(T+1)}{2}$  possible mentions, where  $T$  is the number of words in a document  $D$ .

$P(y_i|D)$  is estimated from two scores: The first is the mention score of both mention  $i$  and antecedent  $y_i$ , which express the likelihood of a span being a valid mention. The second is the binary antecedent score of  $y_i$  being an antecedent of mention  $i$ .

All three scores are summed together as follows:

$$s(i, y_i) = \begin{cases} 0, & y_i = \epsilon, \\ s_m(i) + s_m(y_i) + s_a(i, y_i), & y_i \neq \epsilon, \end{cases} \quad (1)$$

where  $\epsilon$  is an empty (also referred as dummy) antecedent. If a mention candidate is aligned to the dummy antecedent and it is not an antecedent of any other mention (so the mention should only be in the cluster with the dummy antecedent), it is not considered a valid mention. As a result, every valid mention has to be in a cluster with at least one other mention, so the model cannot predict singletons.

The scores are then normalized with the softmax function to estimate the probability:

$$P(y_i|D) = \frac{\exp(s(i, y_i))}{\sum_{y' \in Y(i)} \exp(s(i, y'))}. \quad (2)$$

The scores are estimated by FFNN with one output neural without output activation as follows:

$$s_m(i) = FFNN_m(g_i), \quad (3)$$

where  $g_i$  is a vector representation of span  $i$  obtained from text encoder (originally LSTM).

$$s_a(i, j) = FFNN_a([g_i, g_j, g_i \odot g_j, \Phi]), \quad (4)$$

where  $\Phi$  represents pairwise features like distance between spans.

Vector representation  $g_i$  of a span  $i$  is obtained from a concatenation of various features as follows:

$$g_i = [x_{start(i)}, x_{end(i)}, x_i, \Phi], \quad (5)$$

where  $x$  is the whole input and  $x_{start(i)}$ ,  $x_{end(i)}$  are start and end token respectively.  $x_i$  is the sum of all the span tokens weighted with the attention mechanism [6].  $\Phi$  is an additional feature; span length.

In the training phase, we iterate all mentions predicted by a model and maximize the marginal probability of all its correct antecedents as shown in the following equation:

$$J(D) = \log \prod_{i=1}^N \sum_{\hat{y} \in Y(i) \cap \text{GOLD}(i)} P(\hat{y}), \quad (6)$$

where  $\text{GOLD}(i)$  is the set of all gold antecedents of mention  $i$  (in the training data annotation).

Theoretically, we should go over all possible mentions, but in practice, the mentions are prefiltered based on a unary mention score to reduce computational complexity.

The schema of the model is shown in Figure 1.

This end-to-end model has proven to be highly successful, especially on datasets where singletons are not annotated (e.g., CoNLL 2012/OntoNotes), a scenario where traditional two-stage models tend to struggle. The model was later extended by [7] to the higher-order coreference resolution (CR) through iterative refining of span representations with gated attention mechanism. They further optimize pro model speed by pruning the mentions with simple scorer at the first step and rescoring only top  $k$  spans with the precise scorer in the second step. Formally, they add the coarse coreference score:  $S_c(i, j) = g_i \cdot w_c \cdot g_j$ , which is used for antecedent pruning

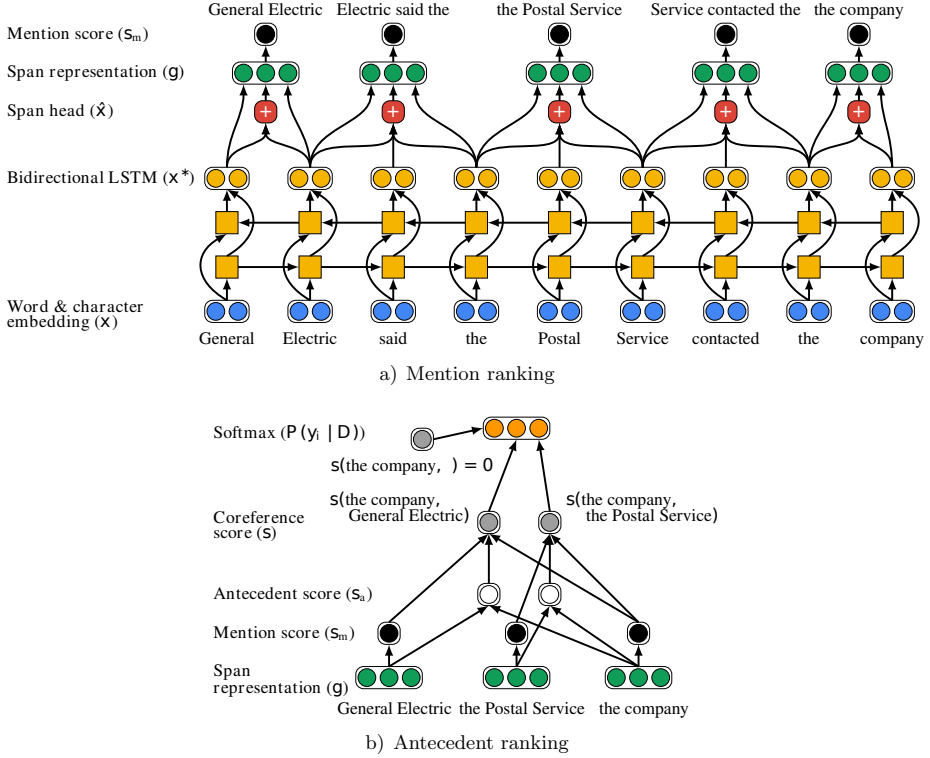


Figure 1. Architecture of the model from [5]

before computing the fine coreference score  $s_a$ . The method is called coarse-to-fine (c2f) coreference resolution. Authors of [8] use the model from [7] but with BERT as an encoder.

While many higher-order CR approaches have been proposed, authors of [3] demonstrated that their impact is marginal when a strong encoder is used, as seen in their experiments with SpanBERT [9].

## 2.2 Word-Level Coreference Resolution

Authors of [4] proposed a word-level coreference resolution approach based on the model by [5]. Their method reduces the computational complexity by reducing the mention space. Instead of iterating over all possible spans, they first map each gold mention to a single word – specifically, the headword, which they select based on the syntax tree (as the node closest to the root of the tree). Antecedent prediction is then performed on the word level. In the next step, they use a span extraction model to predict the original spans from headwords. This model is trained to classify each word in the sentence together with a headword to decide whether the word is

the start token or end token of the span corresponding to the headword. During training, the span extraction model is trained simultaneously with the antecedent prediction as another classification head. The rest of the model is standard coarse-to-fine coreference resolution.

Later in [10], the authors proposed a modification in headword selection, suggesting that when a coordinate conjunction appears within an entity, it should be made the head to resolve span ambiguity. For example, in the sentence “Tom and Mary are playing”, the entities “Tom” and “Tom, and Mary” would typically share the same head in the original approach. Their modification eliminates this ambiguity.

### 2.3 CRAC Shared Task on Multilingual Coreference Resolution

CRAC Shared Task on Multilingual Coreference Resolution (CRAC-coref) [11] is an annual shared task that began in 2022 and is built upon the CorefUD collection. In fact, this paper is an extension of CRAC22-coref [12] participant system [13], which is based on [14].

The best-performing system in CRAC 2022 was submitted by [15]. It is a two-stage model with a joint BERT-like encoder. In the mention detection stage, they use extended BIO<sup>1</sup> tagging via stack manipulation instructions to solve overlapping mentions. Coreference resolution is then done by antecedent prediction head, a component responsible for identifying the most likely antecedent of each mention, which is very similar to the one used in end-to-end neural CR [5]. They use a concatenation of start and end tokens as mention representation and classify mention pairs to find the best antecedent of each mention. The schema of the CorPipe model is shown in Figure 2.

## 3 DATASETS AND METRICS

For the evaluation of the proposed model, we use the CorefUD 1.1 dataset. The collection consists of 17 datasets in 12 different languages. All datasets in the collection are converted into a harmonized form.

The primary goal of CorefUD is to have as many datasets as possible in the same annotation scheme and enable cross-lingual training. Several important design choices have been made by the authors of CorefUD:

- As the collection is the extension of Universal Dependencies [2], all the datasets in the CorefUD collection are treebanks. Some of them were annotated in a treebank form (e.g., Czech-PDT), and the others were parsed during the harmonization procedure.
- Coreference entities are represented as clusters of mentions.

---

<sup>1</sup> BIO stands for Beginning, Inside, and Outside. It is a tagging format to label each token of a text as being the beginning (B), inside (I), or outside (O) of an entity.

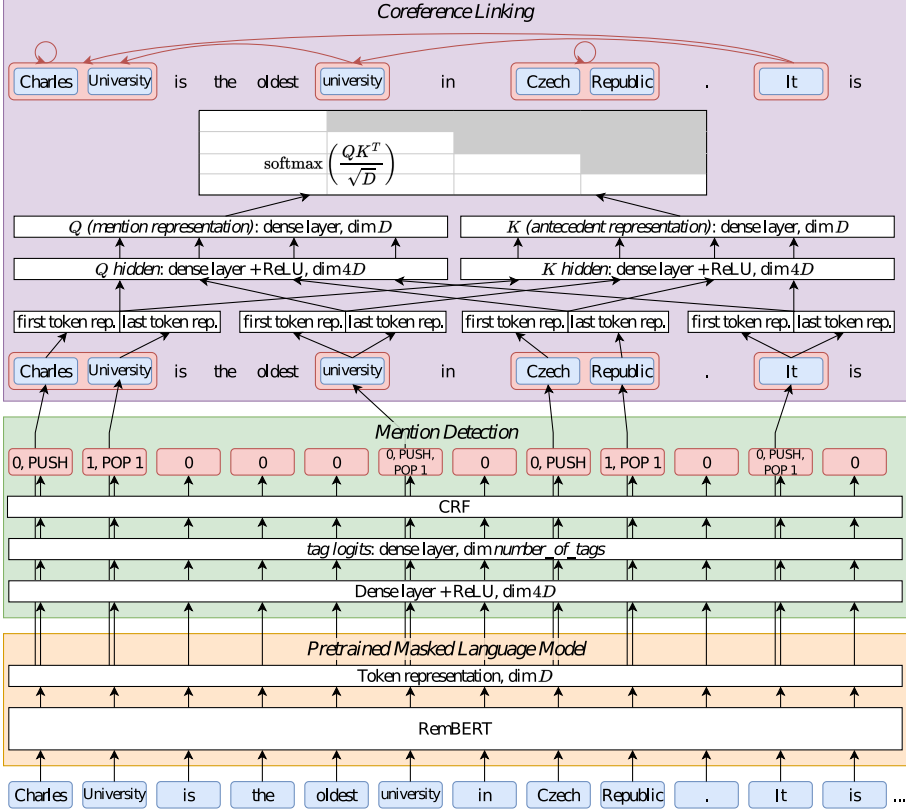


Figure 2. Schema of CorPipe model (from [15])

As highlighted in [1], significant differences remain among individual datasets within CorefUD, not only in data distribution but also in annotation practices. In this section, we discuss the differences that are particularly relevant to our model and experiments.

**Zero mentions and empty nodes:** Some datasets contain zero mentions nodes.

These are mentions that are not explicitly mentioned in the text (cannot be mapped to a word or phrase), but they are implicitly referred to. In CorefUD, zero mentions are captured in empty nodes; virtual nodes are added to a syntactic tree during the annotation process, and they are not mapped to any word. Adding empty nodes as some special words can make it hard for a general-purpose pre-trained model to interpret the meaning of the text accurately. We cannot discard them from the datasets because most of them are part of a coreference cluster. The number of empty nodes in each dataset is shown in Table 1, and the number of zero mentions can be found in Table 3.

**Singletons:** Another critical difference between the datasets lies in the annotation of singletons. Singletons are entities with only a single mention throughout the document, so these mentions have no coreference relations. In some datasets, singletons are annotated, while in others, they are not. That means that when there is an entity with only a single mention, it is not considered an entity during annotation. Table 2 provides the percentage of singleton entities across the datasets (entities with a length of 1).

The CorefUD project is still evolving, and every year, there are some new datasets. The data is also moving towards a realistic scenario. In the very last version, 1.2, test data are practically just plain text (parsed with UDPipe, but no manual annotation is provided).

CorefUD Dataset	Total Size			Division [%]			
	Docs	Sents	Words	Empty Nodes	Train	Dev	Test
ca_ancora	1 298	13 613	429 313	6 377	77.5	11.4	11.1
cs_pcedt	2 312	49 208	1 155 755	35 844	80.9	14.2	4.9
cs_pdt	3 165	49 428	834 720	22 389	78.3	10.6	11.1
en_gum	195	10 761	187 416	99	78.9	10.5	10.6
en_parcor	19	543	10 798	0	81.2	10.7	8.1
fr_democrat	126	13 057	284 883	0	80.1	9.9	10.0
de_parcor	19	543	10 602	0	81.6	10.4	8.1
de_potsdam	176	2 238	33 222	0	80.3	10.2	9.5
hu_korkor	94	1 351	24 568	1 988	79.2	10.3	10.5
hu_szeged	400	8 820	123 968	4 857	81.1	9.6	9.3
lt_lcc	100	1 714	37 014	0	81.3	9.1	9.6
no_bokmaal	346	15 742	245 515	0	82.8	8.8	8.4
no_nynorsk	394	12 481	206 660	0	83.6	8.7	7.7
pl_pcc	1 828	35 874	538 885	470	80.1	10.0	9.9
ru_rucor	181	9 035	156 636	0	78.9	13.5	7.6
es_ancora	1 356	14 159	458 418	8 112	80.0	10.0	10.0
tr_itcc	24	4 733	55 341	0	81.5	8.8	9.7

Table 1. Basic dataset statistics (taken from [1])

### 3.1 Evaluation Metrics

For evaluation, we employ the official metric for CRAC 2023 Multilingual Coreference Resolution shared task implemented via the CoreUD scorer<sup>2</sup>. The metric is a modification of the CoNLL score for coreference resolution, averaging three F1 metrics: *CEAF-e*,  $B^3$ , and *MUC*. Each of these three metrics has a different universe. *MUC* operates on the level of coreference links,  $B^3$  works with mentions and *CEAF-e* is entity-based metric. All these three metrics are described later in

<sup>2</sup> <https://github.com/ufal/corefud-scorer>

CorefUD Dataset	Entities				Distribution of Lengths				
	Total	Per 1k	Length		1	2	3	4	5+
	Count	Words	Max	Avg.	[%]	[%]	[%]	[%]	[%]
ca_ancora	18 030	42	101	3.5	2.6	54.0	18.1	8.6	16.8
cs_pcedt	52 721	46	236	3.3	6.6	59.6	14.7	6.3	12.7
cs_pdt	78 747	94	175	2.4	40.8	35.1	10.4	4.9	8.9
en_gum	27 757	148	131	1.9	73.8	14.2	4.9	2.2	4.9
en_parcor	202	19	38	4.2	6.9	54.0	13.9	5.9	19.3
fr_democrat	39 023	137	895	2.0	81.6	10.7	3.0	1.4	3.3
de_parcor	259	24	43	3.5	6.2	64.9	11.6	5.0	12.4
de_potsdam	3 752	113	15	1.4	76.5	13.9	5.0	1.8	2.7
hu_korkor	1 134	46	41	3.6	0.9	55.1	17.1	9.0	17.9
hu_szeged	5 182	42	36	3.0	8.0	51.1	19.0	9.1	12.9
lt_lcc	1 224	33	23	3.7	11.2	45.3	11.8	8.2	23.5
no_bokmaal	53 357	217	298	1.4	89.4	5.4	1.9	1.0	2.4
no_nynorsk	44 847	217	84	1.4	88.7	5.5	2.1	1.1	2.7
pl_pcc	127 688	237	135	1.5	82.6	9.8	2.9	1.4	3.2
ru_rucor	3 636	23	141	4.5	3.3	53.7	15.6	6.9	20.5
es_ancora	20 115	44	110	3.5	3.3	53.7	17.0	8.8	17.1
tr_itcc	690	12	66	5.3	0.0	39.0	19.6	10.7	30.7

Table 2. Entity statistics (taken from [1]). The entity length means the number of its mentions.

this section. The main modification of the CorefUD scorer lies in using the head matching. In head matching, the gold and system mentions are considered identical if and only if they have the same syntactic head. Another important aspect is that the primary metric ignores singletons.

### 3.1.1 MUC Score

The *MUC* score [16] is a link-based metric. It measures precision and recall of the identification of coreference links in a coreference chain. Coreference chains are formed by linking each mention in a cluster to its antecedent, creating a sequence of references. For instance, precision is computed by summing all correctly predicted links divided by all the links in the system response. In recall, the numerator is the number of all the gold links. Then, we compute the standard F1 measure as the final metric. The number of correctly identified links is technically computed from a partition function  $\mathcal{P}(G_i, R)$ , which determines how many entities cover all the mentions of gold entity  $G_i$ . Recall is then computed as:

$$\text{Recall}_{\text{MUC}} = \frac{\sum_i |G_i| - |\mathcal{P}(G_i, R)|}{\sum_i |G_i| - 1}. \quad (7)$$

CorefUD Dataset	Mentions				Distribution of Lengths						Mention Type		
	Total	per 1k	Length		0	1	2	3	4	5+	w/ empty	w/ gap	non- tree
	Count	Words	Max	Avg.	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
ca_ancora	62417	145	141	4.8	10.2	28.2	21.7	7.9	5.3	26.8	12.4	0.0	3.7
cs_pcedt	168138	145	79	3.6	19.5	29.9	16.8	8.6	4.1	20.9	25.7	0.8	8.4
cs_pdt	154983	186	99	3.1	10.7	39.6	20.3	9.1	4.2	16.1	13.7	1.3	2.2
en_gum	32323	172	95	2.6	0.0	56.5	19.5	8.0	3.8	12.1	0.0	0.0	1.2
en_parcor	835	77	37	2.1	0.0	59.6	23.4	6.3	3.4	7.3	0.0	0.6	0.6
fr_democrat	46487	163	71	1.7	0.0	64.2	21.8	6.3	2.4	5.3	0.0	0.0	2.0
de_parcor	896	85	30	2.0	0.0	64.8	17.5	6.2	4.0	7.4	0.0	0.3	1.5
de_potsdam	2519	76	34	2.6	0.0	34.8	32.4	15.6	6.4	10.9	0.0	6.3	3.8
hu_korkor	4103	167	42	2.2	30.8	20.6	23.2	9.8	4.8	10.8	35.3	0.6	5.5
hu_szeged	15165	122	36	1.6	15.1	37.4	32.5	10.2	2.6	2.2	15.2	0.4	1.1
lt_lcc	4337	117	19	1.5	0.0	69.1	16.6	11.1	1.2	2.0	0.0	0.0	4.3
no_bokmaal	26611	108	51	1.9	0.0	74.5	9.7	6.1	2.1	7.6	0.0	0.6	1.5
no_nynorsk	21847	106	57	2.1	0.0	70.2	10.1	7.7	2.8	9.2	0.0	0.4	1.4
pl_pcc	82804	154	108	2.1	0.3	68.7	14.9	5.2	2.7	8.2	0.5	1.0	4.7
ru_rucor	16193	103	18	1.7	0.0	69.1	16.3	6.6	3.5	4.6	0.0	0.5	1.4
es_ancora	70663	154	101	4.8	11.4	31.6	18.8	7.2	4.5	26.3	14.0	0.0	0.3
tr_itcc	3668	66	25	1.9	0.0	67.3	17.5	5.8	2.9	6.6	0.0	0.0	1.9

Table 3. Non-singleton mention statistics (taken from [1]). The mention length is the number of its non-empty nodes; thus, mentions with the length 0 contain only empty nodes. *w/empty* are mentions where at least one empty node is present, e.g., zero mentions. *w/gap* are mentions with at least one gap, so these are mentions which cannot be represented with a single span. *non-tree* are mentions which do not form a single subtree of the syntax tree.

### 3.1.2 B<sup>3</sup>

$B^3$  [17] was proposed to address problems of *MUC* score. One of them is that *MUC* score as a link-based metric does not address singletons at all.  $B^3$  operates on mention level. It computes precision and recall for every mention and then averages these values. The recall and precision of a mention  $m$  is defined as follows:

$$R_m = \frac{|G_i \cap S_j|}{|G_i|}, \quad (8)$$

$$P_m = \frac{|G_i \cap S_j|}{|S_i|}. \quad (9)$$

### 3.1.3 CEAF

The problem with  $B^3$  is that each entity is addressed many times (with each mention). CEAF [18] tries to solve this problem. It aligns gold and system entities

one-to-one, maximizing the similarity of aligned entities. This leads to the maximum bipartite matching problem. Precision and recall are then defined as follows:

$$P = \frac{\Phi(g^*)}{\sum_i \phi(S_i, S_i)}, \quad (10)$$

where  $\Phi(g^*)$  is sum of similarities of all the matching entities in one-to-one mapping found by Kuhn-Munkres Algorithm, and  $\phi(x, x)$  is self-similarity of an entity.

Author of [18] presents two entity-pair similarity measures that can be used in *CEAF*: one is the absolute number of common mentions between two entities, and the other is a “local” mention F-measure between two entities. The two measures lead to the mention-based *CEAF-m* and entity-based *CEAF-e*, respectively [19].

## 4 BASELINE MODEL AND EXTENSIONS

We adopt an end-to-end model from [3], and we use it as the baseline model (without higher-order inference). We use UDAPI<sup>3</sup> to load the data in CorefUD format. We use XLM-Roberta-large as an encoder.

Building on this baseline, we propose several extensions to the standard end-to-end model presented earlier in this paper. Our objective is to create a universal model suitable for all datasets in the CorefUD collection. We also reevaluate some extensions proposed by [13] for comparison (namely syntax representation and *Span2head* model).

### 4.1 Cross-Lingual Training

Some of the datasets are relatively small (see Table 1). To enhance performance, we propose pretraining the model on a concatenation of all training datasets within the CorefUD collection.

Model	Pretrained Params	New Params
mBERT	180 M	15 M
XLM-R	550 M	20 M

Table 4. Number of trainable parameters of the models

Table 4 shows the total number of parameters and the number of parameters trained from scratch. We can see that XLM-R large has approximately 20 million parameters trained from scratch<sup>4</sup>. Training so many parameters from scratch can lead to overfitting, especially for smaller datasets. To mitigate this, we first train a joint model on the concatenation of all the datasets and then we finetune this model on each dataset separately.

<sup>3</sup> <https://github.com/udapi/udapi-python>

<sup>4</sup> The number is the average of different configurations since different extensions add different numbers of parameters.

## 4.2 Syntactic Information

We believe that integrating syntactic information can greatly improve the performance. Dependency information is crucial for identifying mention heads, as some datasets in the CorefUD collection label mentions as nodes within the dependency tree rather than spans in the linear text sequence.

We use the syntactic information model from [13]. In this model, we modify a model to encode syntactic information in the following way: For each token, we add its path to *ROOT* to its embedding representation. To encode a path in a dependency tree, we use a concatenation of BERT embeddings of all the nodes on the path together with the learned embedding of corresponding dependency relations. To have a fixed-length word representation, we set a maximum tree-path length. By adding tree representation, we increase the total size of a token representation in the model by  $max\_tree\_depth \times (bert\_emb\_size + deprel\_emb\_size)$ . In our experiments, we use the maximum tree-path length of 5.

## 4.3 Span2Head Model

Since the official evaluation uses a head-match metric, we trained the model to predict the heads of mentions rather than full spans to better align with the evaluation criterion. With access to all the necessary information, including dependency trees, the model should be able to infer the original rules for selecting heads.

Our goal is to represent mention with a headword only. However, multiple mentions can share the same head. To address this issue, *Span2head* model represents mentions by their full spans and predicts the head of each mention at the top of the CR model, outputting only the headword(s). This ensures that, during cluster formation, mentions are represented by their spans, preventing incorrect merging of clusters for distinct mentions that share a headword.

Authors of [13] proposed two versions of the head prediction model, both implemented as separate classification layers on top of our coreference resolution model. The first *multi-class* model predicts the relative position of the headword(s) within a span, using the span’s hidden representation from the coreference model. Head position probabilities are calculated with sigmoid activation, allowing the model to predict multiple headwords, even though only one is present in the gold data. This serves to optimize the evaluation metric by outputting all likely headword candidates. The schema of the model is shown in Figure 3 a).

The second *binary* model employs binary classification for each span and head candidate pair, which also enables the prediction of multiple headwords for a single span. The schema of the model is shown in Figure 3 b).

## 4.4 Head Representations

Inspired by the word-level coreference resolution model, we decided to try to model mentions only with their syntactic head. As mentioned earlier, this model merges

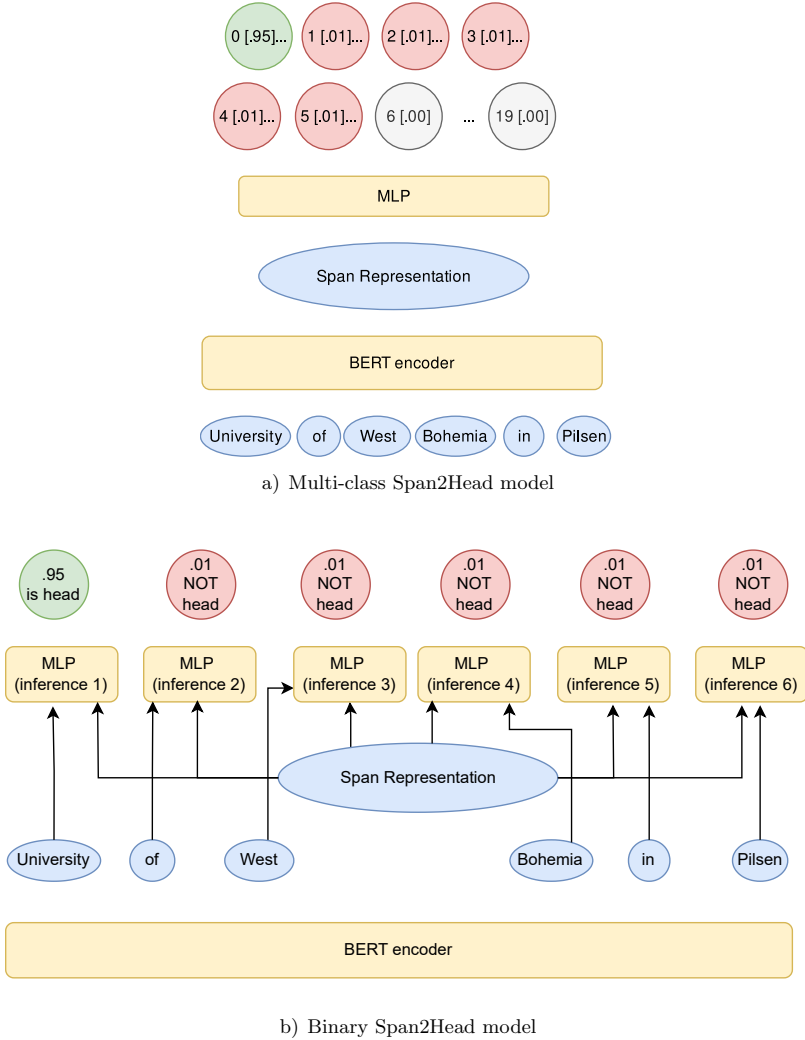


Figure 3. Different Span2Head models

some mentions into one in cases where multiple mentions share the same head. However, according to [4], reduction to headword is highly beneficial. As also mentioned above, for the CRAC official evaluation metric, reconstructing the original span is unnecessary, as predicting the correct head suffices. Therefore, we explored a simplified word-level coreference resolution approach that bypasses the span prediction step. Given that syntactic information is available in all CorefUD datasets, this is not an unrealistic scenario because, in most cases, the spans can be reconstructed by simply taking the whole subtree of the predicted node (head). Heads

are heuristically selected only in cases where a mention does not form a single subtree. The proportion of such mentions is shown in Table 3 (column *non-tree*). Using a word-level model reduces the mention space from quadratic to linear, making the model more efficient and minimizing potential false-positive mentions. Moreover, we believe that for very long mentions, the standard representation (sum of the start token, end token, and attended sum of all tokens) becomes insufficient.

## 4.5 Singletons

Some datasets in the CorefUD collection have singletons annotated, and others do not. Specifically, in CorefUD 1.1, 8 out of 17 datasets have more than 10% singletons, and 6 of these have more than 70% singletons, which is probably a sign of consistent entity annotation independent of the coreference annotation. The baseline model completely ignores singletons during training<sup>5</sup>. As a result, for these 6 singleton-including datasets, we discard more than 70% of training data for the mention identification task. To leverage this data, we incorporate singleton modeling into our model and propose several approaches to address this.

### 4.5.1 Another Dummy Antecedent

In the first proposed approach, we introduce another virtual antecedent (with trained embedding) representing that the mention has no real antecedent, but it is a valid mention, so it is a singleton. For these mentions, we avoid using the binary score because it would unnecessarily model the similarity between all the singletons<sup>6</sup>. Additionally, we include an option to use a separate feed-forward network for computing singleton scores (different from the one for other mentions scores but trained simultaneously on the top of the same encoder).

### 4.5.2 Mention Modeling

The second approach modifies the loss function to model mentions independently of coreference relations. In this approach, we simply add a binary cross-entropy of each span being a mention to the loss function. In other words, we add another classification head for the mention classification:

$$J(D) = \log \prod_{i=1}^N \sum_{\hat{y} \in Y(i) \cap \text{GOLD}(i)} P(\hat{y}) + \underbrace{y_m^{(i)} \cdot \sigma(s_m(i)) + (1 - y_m^{(i)}) \cdot \sigma(-s_m(i))}_{\text{singletons binary cross-entropy}}, \quad (11)$$

<sup>5</sup> The loss is the sum over all correct antecedents, and since singletons have no gold antecedents, they do not affect the loss.

<sup>6</sup> because all the singleton representations would be trained to be similar to the dummy antecedent

where  $y_m^{(i)}$  is 1 if span  $i$  corresponds to gold mention, 0 otherwise.

In the prediction step, the mention score is evaluated only for potential singletons. If a mention has no real antecedent, we look at the mention score. If it is likely to be a mention, we make it a singleton; otherwise, it is not a mention at all.

#### 4.6 Overlapping Segments

One limitation of the XLM-R model is the short sequence length. In the employed model, the input document is split into segments, which are processed individually with a BERT-like encoder and merged in the antecedent-prediction head. To propagate errors correctly across segments, we need to process all segments in a single gradient update. However, due to GPU memory constraints, we set a maximum number of segments, and if a document exceeds this limit, we split it into multiple individual documents without any mutual coreferences.

We propose using segment overlapping with a cluster merging algorithm to address this issue. The cluster merging algorithm is straightforward: We iterate over all mentions in the new part, and if they are present in a cluster in the previous part, we simply make a union between its clusters from both parts. We experiment with minimal (just one overlapping segment) and maximal (each continuing example has just one new segment, the rest are overlapping) segment overlap. For instance, if a document has 6 segments and the maximum number of segments for a single example is 4, then we split the document into 3 examples overlapping with 3 segments. The example is shown in Figure 4. We further extend this method by only using clusters in the continuing examples where at least one mention is present in new segments. If there is no such mention, then all the mentions were seen already in the previous part with a longer left context.

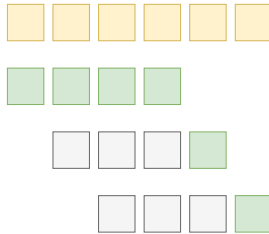


Figure 4. Example of overlapping segments splitting

## 5 EXPERIMENTS

### 5.1 Training

We trained our models using NVIDIA A40 GPUs. The batch size was equal to one document. We limit the maximum sequence length to 6 non-overlapping segments of 512 tokens. During training, if a document exceeded this length, a random offset was used to select a continuous block of 6 segments, with the remaining segments discarded. During prediction, longer documents are split into independent sub-documents. In the basic model, there is no overlap between sub-documents. In the case of the head-only model, we increased the number of segments to eight since this model is more memory-efficient, allowing for a larger sequence length. Each model was trained on its respective dataset for around 80 000 updates in our monolingual experiments. For the models pre-trained on multiple datasets, we used 80 000 steps for joint pre-training across all datasets, followed by roughly 30 000 steps for fine-tuning on each individual dataset.

### 5.2 Baseline and Extensions

We begin by training a monolingual model for each language. For these experiments, we use XLM-Roberta-large and specific monolingual models for each language. Monolingual models for individual languages are listed in Table 5. Then, we train a joint cross-lingual model for all the datasets, which we fine-tune for each dataset afterward.

To evaluate the effect of each extension, we first run all its variants described in Section 4 on the basic XLM-R-large model and measure the performance for each language (Sections 6.1.2 and 6.1.3). Then, we compare each extension in its best configuration to the baseline joint model in Section 6.1.5 (since joint models achieve the best results from our three baseline models). In the last step, we evaluate the effects of Overlapping segments only on the best model configuration for each language (because it is orthogonal to all other extensions).

#### 5.2.1 Monolingual Models

We evaluate a specific monolingual model for each language. We use a large variant where it is available. For Czech we use *Czert* [20], for English *Roberta-large* [21], *gbert-large* [22] for German. The complete list can be found in Table 5.

### 5.3 Zero-Shot Cross-Lingual Experiments

We also conduct zero-shot cross-lingual experiments in two variants: dataset zero-shot and language zero-shot. In dataset zero shot, we train the model on all the datasets except one, and then we evaluate the model on the excluded dataset. However, this approach is not a true cross-lingual zero-shot test, as multiple datasets

exist for some languages, meaning the model was still exposed to the language of the evaluation data during training.

To address this, we performed language zero-shot experiments, where we removed all datasets for a particular language from the training data. Then, we evaluated the model on all datasets in that language. Additionally, we conducted a final experiment focusing on the *ParCor* corpora, which consists of parallel English and German datasets. This allowed us to perform a true zero-shot evaluation on both *ParCor* corpora.

## 6 RESULTS AND DISCUSSION

Table 5 presents the results of baseline models on development datasets parts. Several key observations can be drawn from these results. Most notably, monolingual models tend to outperform the cross-lingual XLM-Roberta model, especially for the Lithuanian and Czech PCEDT datasets, where the monolingual models – even though they are smaller – exhibit significantly better performance. The reason is probably in the difference of these two datasets from the other ones. We can see that for these two datasets, joint training helps only slightly compared to other datasets. For the rest of the datasets, joint training surpasses both other models (monolingual and multilingual trained from scratch).

### 6.1 Effects of Extensions

#### 6.1.1 Joint Training

We further investigate the effect of joint training on the Czech PDT dataset by analyzing concrete documents. We found that the joint model is much better at identifying mentions. For example, if we look at the following part of a document with the gold annotations:

**GOLD** || *Visegrádská<sub>e2</sub> trojka*||<sub>e1</sub> zřejmě nebude mít *volný obchod<sub>e3</sub>*. ||| *Maďarsko<sub>e6</sub> a Polsko<sub>e7</sub>*||<sub>e5</sub> se rozhodly ustoupit od chystané zóny ||*volného obchodu mezi zeměmi* || *visegrádské<sub>e2</sub> trojky*||<sub>e1</sub>||<sub>e3</sub>||<sub>e4</sub>

**GOLD in English** || *The Visegrad<sub>e2</sub> Three*||<sub>e1</sub> will probably not have *free trade<sub>e3</sub>*. ||| *Hungary<sub>e6</sub> and Poland<sub>e7</sub>*||<sub>e5</sub> have decided to withdraw from the planned ||*free trade zone between* || *Visegrad<sub>e2</sub> Three*||<sub>e1</sub>||<sub>e3</sub>||<sub>e4</sub>

**Baseline** || *Visegrádská<sub>e2</sub> trojka*||<sub>e1</sub> zřejmě nebude mít volný obchod. *Maďarsko<sub>e3</sub> a Polsko* se rozhodly ustoupit od chystané zóny volného obchodu mezi zeměmi || *Visegrádské<sub>e2</sub> trojky*||<sub>e1</sub>.

**Joint** ”|| *Visegrádská<sub>e2</sub> trojka*||<sub>e1</sub> zřejmě nebude mít *volný obchod<sub>e3</sub>*. || *Maďarsko<sub>e6</sub> a Polsko<sub>e7</sub>*||<sub>e5</sub> se rozhodly ustoupit od chystané zóny ||*volného obchodu mezi zeměmi* || *visegrádské<sub>e2</sub> trojky*||<sub>e1</sub>||<sub>e3</sub>”,

Model	Monolingual Model Name	Reference	Monoling	XLM-R	Joint
ca_ancora	roberta-base-ca	[23]	<b>74.23 ± .55</b>	70.51 ± .65	<b>74.42 ± .25</b>
cs_pcedt	Czert-B-base	[20]	<b>73.85 ± .24</b>	72.28 ± .28	73.23 ± .08
cs_pdt	Czert-B-base	[20]	70.29 ± .35	<u>71.87 ± .7</u>	<b>73.47 ± .23</b>
de_parcor	gbert-base	[22]	59.83 ± 2	<u>71.79 ± 3.6</u>	<b>76.57 ± 1.5</b>
de_potsdam	gbert-large	[22]	64.78 ± 1.7	<u>68.98 ± 2.4</u>	<b>76.3 ± 1.1</b>
en_gum	roberta-large	[21]	<u>67.415 ± 2.5</u>	65.73 ± .36	<b>69.04 ± .49</b>
en_parcor	roberta-large	[21]	66.39 ± 7.0	67.66 ± 2.9	69.16 ± 2.0
es_ancora	roberta-large-bne	[23]	70.73 ± .23	<u>73.27 ± .7</u>	<b>76.09 ± .13</b>
fr_democrat	camembert-large	[24]	57.08 ± .9	57.35 ± 1.6	<b>65.24 ± .49</b>
hu_korkor	hubert-base-cc	[25]	60.68 ± 1.6	59.44 ± 1.0	<b>68.99 ± .95</b>
hu_szeged	hubert-base-cc	[25]	66.75 ± 1.7	67.67 ± .44	<b>69.52 ± .59</b>
lt_lcc	Litlat-bert	[26]	<b>76.84 ± 1.1</b>	72.52 ± .48	<u>73.59 ± 1.1</u>
no_bokmaal	norbert3-large	[27]	<u>73.47 ± 1.2</u>	72.11 ± .78	<b>74.79 ± .44</b>
no_nynorsk	norbert3-large	[27]	<u>73.51 ± .67</u>	72.32 ± .55	<b>75.98 ± .17</b>
pl_pcc	Herbert-large	[28]	<b>74.3 ± .29</b>	72.48 ± .61	<b>74.1 ± .23</b>
ru_rucor	Rubert-base	[29]	64.69 ± .65	<u>68.52 ± .35</u>	<b>70.38 ± .64</b>
tr_itcc	Electra.-turkish-dis.		18.39 ± 1.3	<u>21.49 ± 1.6</u>	<b>45.39 ± 1.3</b>
avg			65.48	66.23	70.96

Table 5. Results of baseline models on dev data. CRAC 2023 official evaluation metric. *Monoling* column shows the result of the monolingual model specific to each language. The *XLM-R* column presents results of XLM Roberta large trained separately for each dataset. The *Joint* column corresponds to the joint model described in Section 4.1.

where only the very long entity  $e_4$  is missing. (This is caused by maximum mention length, as discussed later.)

We have also noticed that the most significant difference between the baseline and joint model is in the case of foreign news, where the same topic can appear in multiple datasets (as this news from the European parliament).

### 6.1.2 Span2Head

Table 6 shows the performance of different variants of *Span2head* model. We added *Heads* line with heads-only mention representation to show if it is better to use the Span2head model or the heads-only model (since they are mutually exclusive). From the results, we can make several conclusions:

1. Span2head model slightly helps.
2. The Binary model achieves better results than the multi-class model.
3. For all the datasets where the Span2head model helps, the heads-only model is much better, so it does not make sense to use the Span2head model in the next experiments.

Another interesting point is that for a monolingual model trained from scratch, the heads-only model improves the results on 13 out of 17 datasets, and the results are worse only for a single dataset (English-ParCor), which is very small, so the difference might be caused by a random chance.

S2H Model	ca_ancora	cs_pcedt	cs_pdt	de_parcor	de_potsdam	en_gum	en_parcor	es_ancora	fr_democrat
None	70.7	72.06	71.21	<b>72.99</b>	68.26	64.06	<b>69.65</b>	73.18	55.97
Multi-class	65.08	62.73	66.19	64.44	66.03	61.13	58.4	66.75	50.18
Binary	65.16	69.77	<u>71.64</u>	65.73	<u>70.77</u>	<u>64.64</u>	64.42	65.2	51.3
Heads	<b>78.93</b>	<b>74.83</b>	<b>78.08</b>	<b>73.18</b>	<b>73.28</b>	<b>72.95</b>	67.42	<b>79.7</b>	<b>60.78</b>

S2H Model	hu_korkor	hu_szeged	it_lcc	no_bokmaal	no_nynorsk	pl_pcc	ru_rucor	tr_itcc	Average
None	<b>60.08</b>	68.02	72.48	<b>72.83</b>	72.2	72.32	68.68	20.51	66.19
Multi-class	46.52	63.76	69.5	67.52	67.3	69.97	66.03	17.32	60.52
Binary	58.32	67.44	72.01	72.31	71.34	72.39	69.42	17.08	64.06
Heads	<b>60.34</b>	<b>68.87</b>	<b>75.13</b>	<b>72.93</b>	<b>75.05</b>	<b>74.19</b>	<b>72.05</b>	<b>27.82</b>	69.74

Table 6. Results of XLM-R baseline with different types of Span2Head model. *None* line shows the results of the baseline XLM-R-large model without any extensions.

### 6.1.3 Singletons

Table 7 shows the results of all proposed variants of the singletons extension. *None* column represents the baseline *xlm-r-large* trained from scratch for each dataset. *Dummy* column shows the results of plain dummy antecedent for singletons. *mask* extends the previous method binary score masking for singletons. This means that the method does not use the binary score for similarities to the singleton dummy antecedent. *separate* uses separate FFNN for singletons score prediction (different from the one used for standard mentions). The last method *mentions* represents the method described in Section 4.5.2. Additionally, the table shows the number of parameters added by each variant of the model. The last column shows the percentage of singleton entities in each dataset (datasets with few singletons do not need any singleton model).

The table shows that, on average, the best results were achieved with the mention model. *Mask* model has very poor results. Still, it makes sense because the number of specific parameters for singletons modeling is very low (3092 params are used as embedding for dummy antecedent). If we discard the binary score, we reduce the learning power of the model even more. *Dummy* model does not help for any dataset statistically significantly. *Separate* and *Mentions* models are significantly

Model + Params	None	Dummy	Mask	Separate	Mentions	% of Singletons
	0	3 092	3 092	40 M	40 M	–
ca_ancora	70.7	70.81	68.75	70.82	<b>71.61</b>	2.6
cs_pcedt	<b>72.06</b>	<b>72.03</b>	71.65	71.81	<b>72.44</b>	6.6
cs_pdt	71.21	70.75	70.04	72.23	<b>72.84</b>	40.8
de_parcor	<b>72.99</b>	70.23	70.09	65.47	71.29	6.2
de_potsdam	68.26	67.99	68.24	68.44	67.49	76.5
en_gum	64.06	70.4	69.81	70.89	<b>71.61</b>	73.8
en_parcor	<b>69.65</b>	67.49	64.78	65.89	64.43	6.9
es_ancora	73.18	72.88	70.98	<b>73.74</b>	<b>73.85</b>	3.3
fr_democrat	55.97	55.6	53.96	53.86	<b>59.75</b>	81.6
hu_korkor	<b>60.08</b>	<b>60.56</b>	56.6	55.59	57.91	0.9
hu_szeged	<b>68.02</b>	67.1	66.57	67.1	67.03	8.0
lt_lcc	72.48	72.15	<b>75.59</b>	73.3	74.48	11.2
no_bokmaal	72.83	73.28	73.33	73.84	<b>74.15</b>	89.4
no_nynorsk	72.2	72.99	73.57	<b>74.77</b>	74.26	88.7
pl_pcc	72.32	73.78	72.81	<b>74.4</b>	<b>74.49</b>	82.6
ru_rucor	<b>68.68</b>	67.56	67.91	68.55	<b>69.12</b>	3.3
tr_itcc	<b>20.51</b>	<b>21.11</b>	16.07	20.24	13.21	0.0
avg	66.19	66.38	65.34	65.94	66.47	–

Table 7. Results of XLM-R baseline with different types of singletons model

better for *Spanish* (*Mentions* even for *Catalan*) which has almost no singletons. One reasonable explanation is that the baseline model is under-parametrized for this dataset. We evaluated this hypothesis by increasing the model size, but the improvement was not statistically significant. In the following chapter, we evaluate singletons on the model with joint pretraining; we compute confidence intervals, and in Table 13, we can see that the improvement is below statistical significance (on 95 % significance level). So, we believe that these strange results in Table 7 are caused by random chance.

Another important conclusion from this table is that across all datasets with singletons, the singletons model achieves better results than the baseline. The reason is probably in a better mention identification due to more training data. We also analyzed an additional metric, mention overlap ratio (MOR), which measures token overlap between gold and system mentions. Singleton modeling increased MOR from 65 % to 73 % compared to baseline. We investigated concrete examples to validate this hypothesis. An example from the English GUM dataset follows:

**baseline** *Research on adult-learned second language ( $L2_{e1}$ ) has provided considerable insight into the neurocognitive mechanisms underlying the learning and processing of  $L2_{e1}$  grammar $_{e2}$ .*

**singletons/GOLD** *Research on adult-learned second language<sub>e2</sub> (L<sub>2e2</sub>)<sub>e1</sub> has provided considerable insight into the neurocognitive mechanisms underlying the learning and processing of L<sub>2e2</sub> grammar<sub>e6e5e4e3</sub>.*

We can see that the model with singletons can handle even complicated nested mentions, whereas the baseline cannot.

#### 6.1.4 Headword Representation

We went through some examples on the datasets where heads-only representation helps very significantly. For example, in Czech PDT, we found that there are many very long entity mentions that are not hard to capture with the standard span-based model. Sometimes a mention does not even fit into the maximum span length. We believe this is the main reason why the head-based model performs so much better. For example, in the sentence from the Czech PDT corpus:

“Ústavní soud rozhodl ve středu o podnětu skupiny 19 členů pražských zastupitelstev na vypuštění té části zákona o volbách do obecních zastupitelstev, která podmiňuje registraci sdružení nezávislých kandidátů předložením petice podepsané sedmi procenty obyvatel obce.”

In English:

“On Wednesday, the Constitutional Court ruled on a petition filed by a group of 19 members of the Prague City Council to strike out the part of the law on municipal elections that makes the registration of an association of independent candidates contingent on the submission of a petition signed by seven percent of the municipality’s residents.”

The whole sentence is a single entity representing the concrete decision of the court, and there are many mentions of this decision later in the document.

#### 6.1.5 Best Combinations

Table 8 presents the performance of all the extensions proposed. The table also shows the 95% confidence intervals. *Span2head* and *Only head* extend *Trees* (both also uses syntax tree representations). We can see, that tree representations alone improve the results only for *ParCor* corpora. *Span2Head* model helps significantly only for *en\_gum* and *hu\_korkor* corpora but it is not better than *only heads* model. *Singletons* modeling helps for most of the datasets that have singletons annotated. The exceptions are in the Norwegian Bokmaal and German Potsdam datasets. Another interesting point is that for the German ParCor dataset, singletons modeling helps significantly even though de-parcor itself does not contain singletons. One possible explanation is in the joint model. Adding the singletons from other languages to the training data can help to predict the mentions in the German ParCor dataset. Results in Table 7 support this theory because we can see that when we train the model from scratch, singletons modeling does not help. *Only heads* model helps for

Extension	Joint	Trees	Span2head	Singletons	Only Heads	Best Comb.
ca_ancora	74.42 ± .25	74.17 ± .27	72.65 ± .39	<u>74.93 ± .33</u>	<b>82.22 ± .4</b>	82.12 ± .13
cs_pcedt	73.23 ± .08	73.2 ± .1	71.27 ± .11	<u>73.17 ± .22</u>	<b>75.95 ± .13</b>	75.89 ± .09
cs_pdt	73.47 ± .23	73.2 ± .46	72.98 ± .29	73.37 ± .28	<b>79.43 ± .1</b>	79.51 ± .02
de_parcor	76.57 ± 1.5	<b>79.4 ± 1.1</b>	76.35 ± 1.8	<b>79.57 ± .93</b>	<b>78.63 ± 1.9</b>	79.13 ± .89
de_potsdam	76.3 ± 1.1	<b>78.5 ± 1.6</b>	76.28 ± .94	76.62 ± 1.2	75.16 ± .79	78.41 ± .89
en_gum	69.04 ± .49	<b>70.19 ± .75</b>	<b>70.84 ± .27</b>	<b>71.29 ± .62</b>	<b>75.18 ± .11</b>	<b>75.51 ± .16</b>
en_parcor	69.16 ± 2.0	<b>74.1 ± 2.7</b>	67.19 ± 1.4	<u>70.4 ± 2.3</u>	61.77 ± 1.4	72.8 ± .25
es_ancora	76.09 ± .13	76.07 ± .15	72.97 ± .28	<u>76.2 ± .18</u>	<b>82.39 ± .08</b>	82.43 ± .07
fr_democrat	65.24 ± .49	65.46 ± .4	65.38 ± .49	<b>66.23 ± .44</b>	<b>67.72 ± .81</b>	<b>68.58 ± .23</b>
hu_korkor	68.99 ± .95	67.75 ± 1.5	<b>71 ± .69</b>	<u>69.82 ± 1.5</u>	<b>73.51 ± .86</b>	73.55 ± .57
hu_szeged	69.52 ± .59	69.59 ± .54	69.32 ± .34	<u>69.47 ± .82</u>	<b>70.68 ± .46</b>	70.67 ± .31
lt_lcc	73.59 ± 1.1	<u>74.7 ± .72</u>	73.65 ± .62	<b>76.28 ± .79</b>	<b>76.77 ± .98</b>	<u>77.65 ± .71</u>
no_bokmaal	74.79 ± .44	<u>75.48 ± .49</u>	<u>75.64 ± .66</u>	74.67 ± .24	<b>77.84 ± .29</b>	78.15 ± .29
no_nynorsk	75.98 ± .17	74.79 ± .48	74.96 ± 1	<b>76.59 ± .42</b>	<b>78.51 ± .21</b>	<b>78.81 ± .16</b>
pl_pcc	74.1 ± .23	<u>74.35 ± .13</u>	74.19 ± .24	<b>75.22 ± .2</b>	<b>76.1 ± .11</b>	76 ± .17
ru_rucor	70.38 ± .64	70.23 ± .42	70.49 ± .42	70.6 ± .41	<b>75.8 ± .58</b>	75.98 ± .42
tr_itcc	45.39 ± 1.3	34.79 ± 7.7	29.53 ± .52	45.63 ± .4	45.17 ± 3.9	44.17 ± 1.8
avg	70.95	70.94	69.69	71.77	73.70	74.67

Table 8. Performance gains over the best base model with proposed extensions together with 95% confidence intervals. The models that surpass the baseline significantly are bold. The results, which are better only on lower confidence levels, are underlined. Variants selected for the best models are green.

almost all the datasets since it reduces the mention space and improves precision significantly. The extensions selected for the best model are highlighted in green color and the last column shows the results of the best combination. Please note that the results may show slight inconsistencies, even for datasets where a single extension was selected (e.g., ‘Only heads’ and ‘Best Comb.’ for Hungarian). This is because the experiments were rerun, averaging results across different runs with the same configuration. However, these differences are within the confidence intervals, ensuring the results remain reliable.

### 6.1.6 Overlapping Segments

Table 9 evaluates longer context implementation through overlapping segments. We use a maximum length of 8 segments<sup>7</sup>, and we evaluate minimum and maximum overlap (1 and 7, respectively). We also try variants with filtering already-seen mentions as described in Section 4.6 (columns *min\_filter* and *max\_filter*). For most of the datasets, the results do not change at all, so additionally, we run the same experiment with just 4 segments (Table 10)<sup>8</sup>, and we count basic statistics of long coreference relations (Table 11). Table 11 presents three types of information:

<sup>7</sup> since we use this maximum length in most of our other experiments

<sup>8</sup> which is the minimum number of segments we used in our models

max. seg. 8	None	min	min_filter	max	max_filter
ca_ancora	82.39 ± .25	<u>82.57 ± .25</u>	<u>82.54 ± .24</u>	82.5 ± .24	82.5 ± .25
cs_pcedt	78.23 ± .13	78.18 ± .11	78.23 ± .15	78.19 ± .15	78.12 ± .1
cs_pdt	80.02 ± .21	79.95 ± .19	79.97 ± .2	79.98 ± .2	79.96 ± .19
de_parcor	81.54 ± 1.3	81.54 ± 1.3	81.54 ± 1.3	81.54 ± 1.3	81.54 ± 1.3
de_potsdam	75.88 ± 2.2	75.88 ± 2.2	75.88 ± 2.2	75.88 ± 2.2	75.88 ± 2.2
en_gum	76.25 ± .12	76.25 ± .12	76.25 ± .12	76.25 ± .12	76.25 ± .12
en_parcor	67.92 ± 3.1	67.92 ± 3.1	67.92 ± 3.1	67.92 ± 3.1	67.92 ± 3.1
es_ancora	82.65 ± .27	82.65 ± .27	82.65 ± .27	82.65 ± .27	82.65 ± .27
fr_democrat	<u>69.49 ± .46</u>	68.44 ± .69	69.01 ± .44	69.03 ± .58	68.9 ± .91
fr_split	69.44 ± .42	<u>69.97 ± .15</u>	<u>69.98 ± .20</u>	69.43 ± .50	69.66 ± .48
hu_korkor	73.41 ± .43	73.41 ± .43	73.41 ± .43	73.41 ± .43	73.41 ± .43
hu_szeged	71.2 ± .14	71.2 ± .14	71.2 ± .14	71.2 ± .14	71.2 ± .14
lt_lcc-corefud	76.82 ± .41	76.82 ± .41	76.82 ± .41	76.82 ± .41	76.82 ± .41
no_bokmaal	78.96 ± .37	78.96 ± .37	78.96 ± .37	78.96 ± .37	78.96 ± .37
no_nynorsk	80.5 ± .14	80.5 ± .14	80.5 ± .14	80.5 ± .14	80.5 ± .14
pl_pcc	75.97 ± .17	<u>76.14 ± .23</u>	76.05 ± .22	76.06 ± .23	<u>76.1 ± .15</u>
ru_rucor	75.27 ± .42	<b>77.23 ± .63</b>	76.65 ± .39	76.77 ± .39	<b>77.19 ± .57</b>
tr_itcc	44.32 ± 2.2	45.22 ± 1.5	45.26 ± 1.6	45.22 ± 1.4	45.32 ± .98
avg	74.46	74.60	74.60	74.57	74.60

Table 9. Results of the best model configurations with different approaches to merge segments during the prediction step. Maximum 8 segments.

max. seg. 4	None	min	min filter	max	max filter
ca_ancora	82.39 ± .24	<u>82.52 ± .25</u>	82.5 ± .22	82.42 ± .24	82.21 ± .24
cs_pcedt	77.82 ± .088	77.29 ± .17	77.47 ± .17	77.84 ± .096	77.65 ± .11
cs_pdt	79.9 ± .22	79.87 ± .16	79.84 ± .2	79.91 ± .22	79.74 ± .19
de_parcor	81.54 ± 1.3	81.54 ± 1.3	81.54 ± 1.3	81.54 ± 1.3	81.54 ± 1.3
de_potsdam	75.88 ± 2.2	75.88 ± 2.2	75.88 ± 2.2	75.88 ± 2.2	75.88 ± 2.2
en_gum	76.25 ± .12	76.25 ± .12	76.25 ± .12	76.25 ± .12	76.25 ± .12
en_parcor	67.92 ± 3.1	67.92 ± 3.1	67.92 ± 3.1	67.92 ± 3.1	67.92 ± 3.1
es_ancora	82.59 ± .27	82.48 ± .31	82.56 ± .26	82.59 ± .27	82.57 ± .28
fr_democrat	68.81 ± .84	68.64 ± .89	68.33 ± .79	68.81 ± .92	67.77 ± 1.7
fr_split	69 ± .6	<u>69.46 ± .59</u>	<u>69.48 ± .72</u>	67.83 ± .72	68.75 ± .33
hu_korkor	73.41 ± .43	73.41 ± .43	73.41 ± .43	73.41 ± .43	73.41 ± .43
hu_szeged	71.2 ± .14	71.2 ± .14	71.2 ± .14	71.2 ± .14	71.2 ± .14
lt_lcc	76.82 ± .41	76.82 ± .41	76.82 ± .41	76.82 ± .41	76.82 ± .41
no_bokmaal	78.96 ± .36	78.91 ± .38	78.87 ± .34	78.96 ± .36	78.47 ± .37
no_nynorsk	79.87 ± .57	<b>80.32 ± .18</b>	80.15 ± .37	79.87 ± .57	78.94 ± .12
pl_pcc	75.94 ± .15	76.05 ± .18	76.03 ± .22	75.93 ± .15	75.71 ± .17
ru_rucor	75.58 ± .37	<b>76.54 ± .31</b>	75.89 ± .55	75.62 ± .42	72.8 ± .25
tr_itcc	43.61 ± 1.5	43.85 ± 1.8	43.81 ± 1.3	43.59 ± 1.5	42.19 ± 2.7
avg	74.31	74.39	74.33	74.24	73.88

Table 10. Results with different approaches to merge segments. Maximum 4 segments.

	cross N segment corefs [%]		nearest coref cross N [%]		segments over N [%]	
	4	8	4	8	4	8
ca_ancora	5.975	1.459	.6044	.2747	4.739	1.422
cs_pcedt	15.82	7.398	1.107	.1441	13.55	2.133
cs_pdt	9.787	1.54	.4787	.06769	4.175	.3976
de_parcor	0.0	0.0	0.0	0.0	0.0	0.0
de_potsdam	0.0	0.0	0.0	0.0	0.0	0.0
en_gum	0.0	0.0	0.0	0.0	0.0	0.0
en_parcor	0.0	0.0	0.0	0.0	0.0	0.0
es_ancora	3.012	0.0	.1396	0.0	.5076	0.0
fr_democrat	52.86	25.12	1.664	.8106	34.86/27.52	27.52/12.84
hu_korkor	0.0	0.0	0.0	0.0	0.0	0.0
hu_szeged	0.0	0.0	0.0	0.0	0.0	0.0
lt_lcc	0.0	0.0	0.0	0.0	0.0	0.0
no_bokmaal	9.026	0.0	.1092	0.0	3.846	0.0
no_nynorsk	15.49	0.0	.5405	0.0	10.29	0.0
pl_pcc	11.79	2.724	.3591	.1554	2.954	.8439
ru_rucor	43.77	21.76	3.273	1.283	35.44	16.46
tr_itcc	55.6	25.07	5.703	2.037	55.56	11.11

Table 11. Long documents and distant coreference relations statistics

1. Column *cross N segment corefs* shows the percentage of coreference links that cross the boundary of N segments. These links cannot be predicted without the overlapping segments.
2. Column *nearest coref cross N* is the percentage of mentions for which the nearest antecedent is already in a different segments block, so all the antecedents are in the different block, and the antecedent cannot be predicted with the standard model.
3. Column *segments over N* shows the percentage of segments that are after the first N-segments block. Only those segments beyond the first N-segment block are affected by predictions involving segment overlapping, as the remaining segments fit within the model’s capacity and are processed in a single example.

A significant number of log-distant coreferences are in French, Turkish, and Russian datasets. Some of them are also in both Czech datasets, Catalan and Polish. By far the most long coreferences are in the French dataset. From Table 9, we can see that segments overlapping decreases the performance. We investigated the dataset closely and found out that in this dataset, multiple documents are merged into a single document quite often. When this happens, cross coreferences between those concatenated documents are not annotated so the model which can predict long-distant coreferences is penalized. To evaluate the models correctly, we add the

variant where we split the documents in French dev data manually (row *fr\_split*). After splitting, results with segments overlapping improve significantly, but they are still not as high as we expected in comparison with Russian, where the overlap helps significantly. To evaluate this a bit further, we also computed the recall for cross-segment coreference links. The recall is 72% for Russian and only 44% for French. By comparing these datasets manually, we conclude that the coreference chains are very long in the French dataset, and it is probably too hard to predict them correctly from the antecedents (one mistake in the chain might have a huge effect on the overall performance). For the Turkish dataset, it is hard to make any significant improvement due to its low quality.

## 6.2 Cross-Lingual Transfer

Dataset	Fine-tuned	Joint	Zero-shot	Lang. Zero-shot
ca_ancora	82.12 ± .13	81.9225 ± 0.38	72.01 ± .17	58.53 ± .49
cs_pcedt	<b>75.89 ± .09</b>	74.7775 ± 0.23	62.26 ± .02	51.78 ± .24
cs_pdt	<b>79.51 ± .02</b>	78.8575 ± 0.25	69.92 ± .16	64.55 ± .28
de_parcor	<b>79.13 ± .89</b>	65.095 ± 3.02	56.77 ± .82	<u>62.33 ± .3</u>
de_potsdam	<b>78.41 ± .89</b>	75.1325 ± 2.24	67.15 ± 1.4	64.33 ± .64
en_gum	<b>75.51 ± .16</b>	74.71 ± 0.41	60.02 ± .39	<u>62.03 ± .38</u>
en_parcor	<b>72.8 ± .25</b>	46.3575 ± 1.29	<u>58.06 ± 1.9</u>	46.67 ± .81
es_ancora	82.43 ± .07	82.245 ± 0.35	75.09 ± .26	61.68 ± .38
fr_democrat	<b>68.58 ± .23</b>	67.7225 ± 0.5	60.32 ± .14	–
hu_korkor	73.55 ± .57	73.2625 ± 0.54	58.26 ± .31	53.75 ± .35
hu_szeged	70.67 ± .31	69.75 ± 0.65	55.16 ± .11	53.36 ± .17
lt_lcc	<b>77.65 ± .71</b>	75.65 ± 0.78	47.93 ± .59	–
no_bokmaal	78.15 ± .29	77.8225 ± 0.67	74.83 ± .05	64.77 ± .76
no_nynorsk	<b>78.81 ± .16</b>	77.955 ± 0.46	72.7 ± .37	63.24 ± .94
pl_pcc	76 ± .17	76.0625 ± 0.08	58.44 ± .13	–
ru_rucor	<b>75.98 ± .42</b>	72.665 ± 0.83	62.12 ± .28	–
tr_itcc	44.17 ± 1.8	38.52 ± 1.61	<b>46.26 ± .2</b>	–
avg	74.67	71.09	62.19	58.92

Table 12. Zero-shot cross-lingual transfer results. The table compares four settings. A joint model trained for all datasets, a joint model fine-tuned on each dataset separately afterward, a zero-shot setting for each dataset, and for each language.

Table 12 shows the results of cross-lingual transfer evaluation. We can see that for most of the datasets, fine-tuning the model on the specific data helps, but the difference is not so significant. For several datasets, the results are even comparable. For small datasets, the fine-tuned models achieve much better results than the joint model. Note that we do not use any weighing of particular datasets in the joint collection, so small datasets have a marginal effect on the training. The results of dataset zero-shot and language zero-shot are mostly very predictable. For the

languages with more datasets in CorefUD collection, the drop of zero-shot compared to the joint model was smaller than for the rest of the datasets. When we discard all the datasets for each language, the results are mostly significantly worse compared to discarding just a single dataset. This behavior is expected. Several exceptions should be analyzed more. The strangest results are on the Turkish dataset. Here, the zero-shot model achieves the best performance. Together with the fact that the numbers are very low for this dataset, it suggests that there is some noise in the dataset annotation. If we train the model on the training part of the Turkish dataset the noise is present on two sides (training and testing). If we train a zero-shot model, we eliminate one part. Another interesting aspect is the superior performance of `de_parcor` and `en_gum` datasets in the language zero-shot scenario.

### 6.3 Final Results

	<code>ca_ancora</code>	<code>cs_pcedt</code>	<code>cs_pdt</code>	<code>de_parcor</code>	<code>de_potsdam</code>	<code>en_gum</code>	<code>en_parcor</code>	<code>es_ancora</code>	<code>fr_democrat</code>	<code>hu_korkor</code>
Our Best	<b>82.05</b>	<b>78.85</b>	<b>78.23</b>	<b>77.62</b>	71.07	<b>73.19</b>	<b>61.62</b>	<b>82.71</b>	<b>69.59</b>	<b>69.03</b>
CorPipe	79.93	76.02	76.76	63.3	<b>72.63</b>	72.33	57.58	81.18	65.42	66.18
CorPipe-l	82.39	77.93	77.85	69.94	67.93	75.02	64.79	82.26	68.22	67.95
	<code>hu_szeged</code>	<code>it_lcc</code>	<code>no_bokmaal</code>	<code>no_nynorsk</code>	<code>pl_pcc</code>	<code>ru_rucor</code>	<code>tr_itcc</code>	average	OntoNotes	-
Our Best	<b>69.02</b>	<b>70.69</b>	<b>76.24</b>	<b>74.54</b>	77	<b>81.48</b>	<b>56.45</b>	<b>73.49</b>	80.5	-
CorPipe	65.4	68.63	75.43	73.64	<b>79.04</b>	78.43	42.47	70.26	<b>80.7</b>	-
CorPipe-l	69.16	75.63	78.94	77.24	78.93	80.35	49.97	73.21	82	-
wl-coref	-	-	-	-	-	-	-	-	81	-

Table 13. Final results on test sets. The proposed model compared to two CorPipe variants (without ensembling). CorPipe uses the same encoder as the proposed model (XLNet-large), and CorPipe-large uses a larger T5 model.

Table 13 presents the final results of the best-performing models on a test set. For comparison, we use the best model of CRAC 2023 Multilingual coreference resolution shared task from [30] but not the best submission. The best model is much larger than ours, and it uses ensembling, so as the main model for comparison, we use their RemBERT version without ensembling, which has a comparable size to the proposed model.

We can see that the proposed model outperformed CorPipe on most of the datasets. For some datasets (mostly the ones without singletons annotated), it outperformed even the large CorPipe model, which has approximately 3 times more trainable parameters. The only dataset where CorPipe outperformed our model

by a large margin is the Polish one, which we consider an anomaly because, for this dataset, CorPipe results on the test set are much better than for the dev set. On the dev set, we achieve similar results. The same happened in the opposite way for the Russian dataset. Surprisingly, we even outperformed the large CorPipe model in average score, but this is caused mainly by a large margin on Turkish and German-parcor.

We also evaluate our model on English OntoNotes dataset [31] to validate that our model achieves reasonable performance. It should achieve similar results to word-level coreference since other improvements (like singletons modeling) are not important for OntoNotes. We can see that even though our model is slightly worse than CorPipe, the results are reasonable and close enough to the word-level coreference. We believe that the superior performance of wl-coref can be caused by using specific features for OntoNotes in wl-coref (like speaker information). Note that our goal was not to optimize the results on OntoNotes, so we did not optimize the model specifically for this corpus.

## 7 CONCLUSION

In this paper, we explored and evaluated various approaches to multilingual coreference resolution using the CorefUD 1.1 dataset. Our experiments revealed that monolingual models typically outperform cross-lingual models, especially for languages with datasets that are distinct in their characteristics. However, joint training across languages can still provide benefits for most of the datasets where sufficient cross-linguistic similarities exist.

We proposed several extensions to enhance the baseline models, including cross-lingual training, Span2Head modeling, syntactic information integration, headword mention representation, and long-context prediction. Among these, the heads-only model and singleton modeling showed the most consistent improvements across different datasets, demonstrating the importance of targeted adaptations for coreference resolution tasks. For several datasets, long-context prediction also brings a significant improvement. Additionally, our zero-shot cross-lingual experiments provided insights into the challenges and opportunities of cross-lingual transfer, with the Turkish dataset results highlighting potential issues related to noise in the training data.

Overall, our findings emphasize the need for tailored approaches in coreference resolution, particularly when dealing with diverse languages and annotation schemes. Future work could further investigate the impact of dataset-specific characteristics on model performance and explore additional strategies for enhancing cross-lingual transfer, especially for low-resource languages. Our source codes are publicly available for subsequent research<sup>9</sup>.

---

<sup>9</sup> <https://github.com/ondfa/coref-multiling>

## Acknowledgement

This work has been supported by the project R&D of Technologies for Advanced Digitalization in the Pilsen Metropolitan Area (DigiTech) No. CZ.02.01.01/00/23.021/0008436. Computational resources were supplied by the project “e-Infrastruktura CZ” (e-INFRA LM2018140) provided within the program Projects of Large Research, Development and Innovations Infrastructures.

## REFERENCES

- [1] NEDOLUZHKO, A.—NOVÁK, M.—ŽABOKRTSKÝ, Z.—ZELDES, A.—ZEMAN, D.: CorefUD 1.0: Coreference Meets Universal Dependencies. In: Calzolari, N., Béchet, F., Blache, P. et al. (Eds.): Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC 2022). ELRA, 2022, pp. 4859–4872, <https://aclanthology.org/2022.lrec-1.520/>.
- [2] NIVRE, J.—DE MARNEFFE, M. C.—GINTER, F.—HAJČ, J.—MANNING, C. D.—PYYSALO, S.—SCHUSTER, S.—TYERS, F.—ZEMAN, D.: Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In: Calzolari, N., Béchet, F., Blache, P. et al. (Eds.): Proceedings of the Twelfth Language Resources and Evaluation Conference (LREC 2020). ELRA, 2020, pp. 4034–4043, <https://www.aclweb.org/anthology/2020.lrec-1.497/>.
- [3] XU, L.—CHOI, J. D.: Revealing the Myth of Higher-Order Inference in Coreference Resolution. In: Webber, B., Cohn, T., He, Y., Liu, Y. (Eds.): Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020). ACL, 2020, pp. 8527–8533, doi: 10.18653/v1/2020.emnlp-main.686.
- [4] DOBROVOLSKII, V.: Word-Level Coreference Resolution. In: Moens, M. F., Huang, X., Specia, L., Yih, S. W. (Eds.): Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021). ACL, 2021, pp. 7670–7675, doi: 10.18653/v1/2021.emnlp-main.605.
- [5] LEE, K.—HE, L.—LEWIS, M.—ZETTLEMOYER, L.: End-to-End Neural Coreference Resolution. In: Palmer, M., Hwa, R., Riedel, S. (Eds.): Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017). ACL, 2017, pp. 188–197, doi: 10.18653/v1/D17-1018.
- [6] BAHDANAU, D.—CHO, K.—BENGIO, Y.: Neural Machine Translation by Jointly Learning to Align and Translate. CoRR, 2014, doi: 10.48550/arXiv.1409.0473.
- [7] LEE, K.—HE, L.—ZETTLEMOYER, L.: Higher-Order Coreference Resolution with Coarse-to-Fine Inference. In: Walker, M., Ji, H., Stent, A. (Eds.): Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers) (NAACL 2018). ACL, 2018, pp. 687–692, doi: 10.18653/v1/N18-2108.
- [8] JOSHI, M.—LEVY, O.—ZETTLEMOYER, L.—WELD, D.: BERT for Coreference Resolution: Baselines and Analysis. In: Inui, K., Jiang, J., Ng, V., Wan, X. (Eds.): Proceedings of the 2019 Conference on Empirical Methods in Natural Language Pro-

- cessing and the 9<sup>th</sup> International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019). ACL, 2019, pp. 5803–5808, doi: 10.18653/v1/D19-1588.
- [9] JOSHI, M.—CHEN, D.—LIU, Y.—WELD, D. S.—ZETTLEMOYER, L.—LEVY, O.: SpanBERT: Improving Pre-Training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics*, Vol. 8, 2020, pp. 64–77, doi: 10.1162/tacl.a\_00300.
- [10] D’OOSTERLINCK, K.—BITEW, S. K.—PAPINEAU, B.—POTTS, C.—DEMEESTER, T.—DEVELDER, C.: CAW-Coref: Conjunction-Aware Word-Level Coreference Resolution. In: Ogrodniczuk, M., Ng, V., Pradhan, S., Poesio, M. (Eds.): *Proceedings of the Sixth Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2023)*. ACL, 2023, pp. 8–14, doi: 10.18653/v1/2023.crac-main.2.
- [11] ŽABOKRTSKÝ, Z.—KONOPIK, M.—NEDOLUZHKO, A.—NOVÁK, M.—OGRODNICZUK, M.—POPEL, M.—PRAZAK, O.—SIDO, J.—ZEMAN, D.: Findings of the Second Shared Task on Multilingual Coreference Resolution. In: Žabokrtský, Z., Ogrodniczuk, M. (Eds.): *Proceedings of the CRAC 2023 Shared Task on Multilingual Coreference Resolution*. ACL, 2023, pp. 1–18, doi: 10.18653/v1/2023.crac-sharedtask.1.
- [12] ŽABOKRTSKÝ, Z.—KONOPIK, M.—NEDOLUZHKO, A.—NOVÁK, M.—OGRODNICZUK, M.—POPEL, M.—PRAŽÁK, O.—SIDO, J.—ZEMAN, D.—ZHU, Y.: Findings of the Shared Task on Multilingual Coreference Resolution. In: Žabokrtský, Z., Ogrodniczuk, M. (Eds.): *Proceedings of the CRAC 2022 Shared Task on Multilingual Coreference Resolution*. ACL, 2022, pp. 1–17, <https://aclanthology.org/2022.crac-mcr.1/>.
- [13] PRAŽÁK, O.—KONOPIK, M.: End-to-End Multilingual Coreference Resolution with Mention Head Prediction. In: Žabokrtský, Z., Ogrodniczuk, M. (Eds.): *Proceedings of the CRAC 2022 Shared Task on Multilingual Coreference Resolution*. ACL, 2022, pp. 23–27, doi: 10.26615/978-954-452-072-4.125.
- [14] PRAŽÁK, O.—KONOPIK, M.—SIDO, J.: Multilingual Coreference Resolution with Harmonized Annotations. In: Mitkov, R., Angelova, G. (Eds.): *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*. INCOMA Ltd., 2021, pp. 1119–1123, <https://aclanthology.org/2021.ranlp-1.125/>.
- [15] STRAKA, M.—STRAKOVÁ, J.: ÚFAL CorPipe at CRAC 2022: Effectivity of Multilingual Models for Coreference Resolution. In: Žabokrtský, Z., Ogrodniczuk, M. (Eds.): *Proceedings of the CRAC 2022 Shared Task on Multilingual Coreference Resolution*. ACL, 2022, pp. 28–37, <https://aclanthology.org/2022.crac-mcr.4.>
- [16] VILAIN, M.—BURGER, J. D.—ABERDEEN, J.—CONNOLLY, D.—HIRSCHMAN, L.: A Model-Theoretic Coreference Scoring Scheme. *Proceedings of the 6<sup>th</sup> Conference on Message Understanding (MUC6 ’95)*, 1995, pp. 45–52, doi: 10.3115/1072399.1072405.
- [17] BAGGA, A.—BALDWIN, B.: Algorithms for Scoring Coreference Chains. *The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, Vol. 1, 1998, pp. 563–566.
- [18] LUO, X.: On Coreference Resolution Performance Metrics. In: Mooney, R., Brew, C., Chien, L. F., Kirchhoff, K. (Eds.): *Proceedings of Human Language Technology Con-*

- ference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP 2005). ACL, 2005, pp. 25–32, <https://aclanthology.org/H05-1004/>.
- [19] POESIO, M.—YU, J.—PAUN, S.—ALORAINI, A.—LU, P.—HABER, J.—COKAL, D.: Computational Models of Anaphora. *Annual Review of Linguistics*, Vol. 9, 2023, No. 1, pp. 561–587, doi: 10.1146/annurev-linguistics-031120-111653.
- [20] SIDO, J.—PRAŽÁK, O.—PŘIBÁŇ, P.—PAŠEK, J.—SEJÁK, M.—KONOPÍK, M.: Czert – Czech BERT-Like Model for Language Representation. In: Mitkov, R., Angelova, G. (Eds.): *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*. INCOMA Ltd., 2021, pp. 1326–1338, doi: 10.26615/978-954-452-072-4\_149.
- [21] ZHUANG, L.—WAYNE, L.—YA, S.—JUN, Z.: A Robustly Optimized BERT Pre-Training Approach with Post-Training. In: Li, S., Sun, M., Liu, Y. et al. (Eds.): *Proceedings of the 20<sup>th</sup> Chinese National Conference on Computational Linguistics (CCL 2021)*. Chinese Information Processing Society of China, 2021, pp. 1218–1227, <https://aclanthology.org/2021.ccl-1.108/>.
- [22] CHAN, B.—SCHWETER, S.—MÖLLER, T.: German’s Next Language Model. In: Scott, D., Bel, N., Zong, C. (Eds.): *Proceedings of the 28<sup>th</sup> International Conference on Computational Linguistics (COLING 2020)*. International Committee on Computational Linguistics, 2020, pp. 6788–6796, doi: 10.18653/v1/2020.coling-main.598.
- [23] GUTIÉRREZ FANDIÑO, A.—ARMENGOL ESTAPÉ, J.—PÀMIES, M.—LLOP PALAO, J.—SILVEIRA OCAMPO, J.—PIO CARRINO, C.—ARMENTANO OLLER, C.—RODRIGUEZ PENAGOS, C.—GONZALEZ AGIRRE, A.—VILLEGAS, M.: MarIA: Spanish Language Models. *Procesamiento del Lenguaje Natural*, Vol. 68, 2022, pp. 39–60, doi: 10.26342/2022-68-3.
- [24] MARTIN, L.—MULLER, B.—ORTIZ SUÁREZ, P. J.—DUPONT, Y.—ROMARY, L.—DE LA CLERGERIE, É.—SEDDAH, D.—SAGOT, B.: CamemBERT: A Tasty French Language Model. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (Eds.): *Proceedings of the 58<sup>th</sup> Annual Meeting of the Association for Computational Linguistics (ACL 2020)*. ACL, 2020, pp. 7203–7219, doi: 10.18653/v1/2020.acl-main.645.
- [25] NEMESKEY, D. M.: Introducing huBERT. XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2021), 2021, pp. 3–14.
- [26] ULČAR, M.—ROBNIK-ŠIKONJA, M.: Training Dataset and Dictionary Sizes Matter in BERT Models: The Case of Baltic Languages. Vol. 13217, 2021, pp. 162–172, doi: 10.1007/978-3-031-16500-9\_14.
- [27] SAMUEL, D.—KUTUZOV, A.—TOULEB, S.—VELLDAL, E.—ØVRELID, L.—RØNNINGSTAD, E.—SIGDEL, E.—PALATKINA, A.: NorBench – A Benchmark for Norwegian Language Models. In: Alumäe, T., Fishel, M. (Eds.): *Proceedings of the 24<sup>th</sup> Nordic Conference on Computational Linguistics (NoDaLiDa 2023)*. University of Tartu Library, 2023, pp. 618–633, <https://aclanthology.org/2023.nodalida-1.61>.
- [28] MROCZKOWSKI, R.—RYBAK, P.—WRÓBLEWSKA, A.—GAWLIK, I.: HerBERT: Efficiently Pretrained Transformer-Based Language Model for Polish. In: Babych, B., Kanishcheva, O., Nakov, P. et al. (Eds.): *Proceedings of the 8<sup>th</sup> Workshop on Balto-Slavic Natural Language Processing (BSNLP 2021)*. ACL, 2021, pp. 1–10, <https://www.aclweb.org/anthology/2021.bsnlp-1.1>.

- [29] KURATOV, Y.—ARKHIPOV, M.: Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language. CoRR, 2019, doi: 10.48550/arXiv.1905.07213.
- [30] STRAKA, M.: Úfal CorPipe at CRAC 2023: Larger Context Improves Multilingual Coreference Resolution. In: Žabokrtský, Z., Ogrodniczuk, M. (Eds.): Proceedings of the CRAC 2023 Shared Task on Multilingual Coreference Resolution. ACL, 2023, pp. 41–51, doi: 10.18653/v1/2023.crac-sharedtask.4.
- [31] PRADHAN, S.—MOSCHITTI, A.—XUE, N.—URYUPINA, O.—ZHANG, Y.: CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes. Joint Conference on EMNLP and CoNLL – Shared Task, ACL, 2012, pp. 1–40, <https://www.aclweb.org/anthology/W12-4501>.



**Ondřej PRAŽÁK** is a researcher and Assistant Professor at the Department of Computer Science and Engineering at the University of West Bohemia. The main area of his research is natural language processing, especially cross-lingual semantics. Besides that, he focuses on machine learning methods, especially on deep neural networks. He has been an active researcher since 2015 and has published more than 30 papers. He teaches the Machine Learning, Natural Language Processing, and Information Retrieval courses.



**Miloslav KONOPÍK** is a senior Assistant Professor, AI researcher, and practitioner specializing in NLP. Since 2016, he has served as an academic at the Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia; from 2005 to 2016, he was a researcher at the department and the NTIS Research Centre, and since 2023, he has been running an AI consulting practice. He coordinates and teaches the courses Advanced Natural Language Processing and Applications of Data Types (300+ students) and supervises Ph.D. candidates. His research focuses on semantic understanding of

documents, language models, and multilingual coreference. He has led or co-led numerous nationally and internationally funded projects, all successfully delivered, with results published at leading venues (EMNLP, RANLP, LNCS) and transferred into practice. He co-authored Czert, a Czech BERT-like language model, and the winning system for SemEval 2016 Task 2 (STS). He maintains long-term collaborations with the Institute of Formal and Applied Linguistics (ÚFAL) at Charles University, the Pattern Recognition Lab at FAU Erlangen–Nürnberg, and Ruhr-Universität Bochum, and has conducted contract research for Owen Software, McKinsey, Circularo, and Komerční banka. He has authored 55+ publications with a strong citation record, serves on the program and organizing committees of the TSD conference, and regularly gives invited talks in the Czech Republic and abroad.



**Pavel KRÁL** is Full Professor at the Department of Computer Science and Engineering of the University of West Bohemia (Czech Republic). He is also the head of the NLP research group (<http://nlp.kiv.zcu.cz>) at the NTIS research center of the University of West Bohemia. He received his Ph.D. degree in 2007 from the Department of Computer Science and Engineering at the University of West Bohemia and at Henri Poincaré University in Nancy (France). His research focuses on natural language processing and pattern recognition, with emphasis on semantics, dialogue act recognition, and historical document

analysis. He published more than 100 research papers in prestigious international journals and conferences. He is also responsible for the specialization in Natural Language Processing in the graduate (M.Sc.) study program – Computer Science and its Specialization.