This paper was published at **CRAC 2024** – please cite the published version https://aclanthology.org/2024.crac-1.9.

CorPipe at CRAC 2024: Predicting Zero Mentions from Raw Text

Milan Straka
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské nám. 25, Prague, Czech Republic
straka@ufal.mff.cuni.cz

Abstract

We present CorPipe 24, the winning entry to the CRAC 2024 Shared Task on Multilingual Coreference Resolution. In this third iteration of the shared task, a novel objective is to also predict empty nodes needed for zero coreference mentions (while the empty nodes were given on input in previous years). This way, coreference resolution can be performed on raw text. We evaluate two model variants: a two-stage approach (where the empty nodes are predicted first using a pretrained encoder model and then processed together with sentence words by another pretrained model) and a single-stage approach (where a single pretrained encoder model generates empty nodes, coreference mentions, and coreference links jointly). In both settings, CorPipe surpasses other participants by a large margin of 3.9 and 2.8 percent points, respectively. The source code and the trained model are available at https://github.com/ufal/crac2024-corpipe.

1 Introduction

The CRAC 2024 Shared Task on Multilingual Coreference Resolution (Novák et al., 2024) is a third iteration of a shared task, whose goal is to accelerate research in multilingual coreference resolution (Žabokrtský et al., 2023, 2022). This year, the shared task features 21 datasets in 15 languages from the CorefUD 1.2 collection (Popel et al., 2024).

Compared to the last year—apart from 4 new datasets in 3 languages—a novel task is to predict the so-called *empty nodes* (according to the Universal Dependencies terminology; Nivre et al. 2020). The empty nodes can be considered “slots” that can be part of coreference mentions even if not being present on the surface level of a sentence. The empty nodes are particularly useful in pro-drop languages (like Slavic and Romance languages), where pronouns are sometimes dropped from a sentence when they can be inferred, for example by verb morphology, like in the Czech example *“Řekl, že nepřijde”*, translated as *“(He) said that (he) won’t come”*.

We present CorPipe 24, an improved version of our system submitted in last years (Straka, 2023; Straka and Straková, 2022). We evaluate two variants of the system. In a two-stage variant, the empty nodes are first predicted by a baseline system utilizing a pretrained language encoder model;[1] then, the predicted empty nodes are, together with the input words, processed by original CorPipe using another pretrained encoder. In comparison, a single-stage variant employs a single pretrained encoder model, which predicts the empty nodes, coreference mentions, and coreference links jointly.

Our contributions are as follows:

- We present the winning entry to the CRAC 2024 Shared Task on Multilingual Coreference Resolution, surpassing other participants by a large margin of 3.9 and 2.8 percent points with a two-stage and a single-stage variant, respectively.
- We compare the two-stage and the single-stage settings, showing that the two-stage system outperforms the single-stage system by circa one percent points, both in the regular and the ensembled setting.
- Apart from the CorefUD 1.2, we evaluate the CorPipe performance also on OntoNotes (Pradhan et al., 2013), a frequently used English dataset.
- The CorPipe 24 source code is available at https://github.com/ufal/crac2024-corpipe under an open-source license. The two-stage and the single-stage models are also released, under the CC BY-NC-SA license.

[1] Our implementation of the baseline system was available to all shared task participants in case they do not want to predict the empty nodes themselves.

2 Related Work

Traditionally, coreference resolution was solved by first predicting the coreference mentions and subsequently performing coreference linking (clustering) of the predicted mentions. However, in recent years, the end-to-end approach (Lee et al., 2017, 2018; Joshi et al., 2019, 2020) has become more popular. Indeed, the baseline of the CRAC 2022, 2023, and 2024 shared tasks (Pražák et al., 2021) follow this approach, as well as the second-best solution of CRAC 2022 (Pražák and Konopík, 2022) and the third-best solution of CRAC 2023.

The end-to-end approach has been improved by Kirstain et al. (2021) not to explicitly construct the span representations, and by Dobrovolskii (2021) to consider only the word level, ignoring the span level altogether during coreference linking. Simultaneously, Wu et al. (2020) formulated coreference resolution in a question answering setting, reaching superior results at the expense of substantially more model predictions and additional question-answering data.

The current state-of-the-art results on OntoNotes (Pradhan et al., 2013), a frequently used English coreference resolution dataset, are achieved by autoregressive models with billions of parameters: Liu et al. (2022) propose a specialized autoregressive system, while Bohnet et al. (2023) employ a text-to-text paradigm. However, both these architectures must call the trained model repeatedly to process a single sentence.

3 Two-stage CorPipe

The two-stage variant of CorPipe processes input in two steps: first, empty nodes are predicted using the baseline system available to all shared task participants; then, the coreference resolution is performed using CorPipe. This approach is very similar to the last year's edition of the CRAC Shared Task, where the empty nodes were already given on input. Therefore, the last year's version CorPipe 23 (Straka, 2023) can be used.

3.1 Empty Nodes Baseline

The baseline for predicting empty nodes generates for each empty node only the minimum amount of information needed: the word order position defined by an input word that the empty node should follow (the word order position determines the position of the empty node in coreference mentions) and the dependency head and the dependency relation of the empty node (required by the empty node matching during evaluation); no forms or lemmas are predicted even if provided in the training data. The baseline predicts the empty nodes non-autoregressively, generating at most two empty nodes for every input word; the input word becomes the dependency head of the predicted empty node.

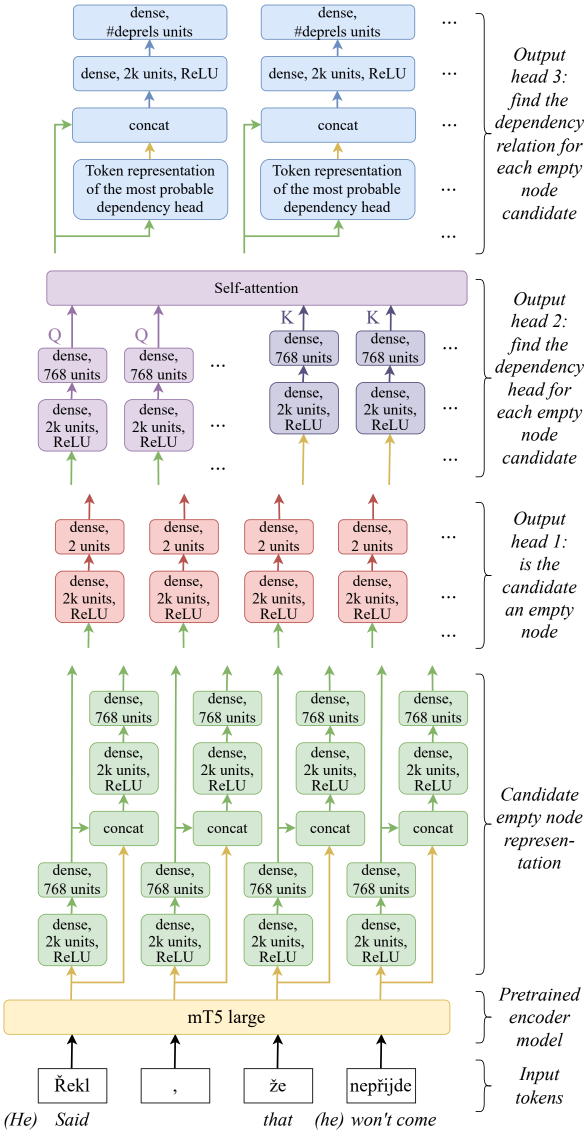
dense, #deprels units
dense, 2k units, ReLU
concat
Token representation of the most probable dependency head
Output head 3: find the dependency relation for each empty node candidate
Self-attention
Q
K
dense, 768 units
dense, 2k units, ReLU
Output head 2: find the dependency head for each empty node candidate
dense, 2 units
dense, 2k units, ReLU
Output head 1: is the candidate an empty node
dense, 768 units
dense, 2k units, ReLU
concat
Candidate empty node represen-tation
mT5 large
Pretrained encoder model
Řekl
,
že
nepřijde
Input tokens
(He) Said
that
(he) won't come

Figure 1: The system architecture of the empty node prediction baseline. Every ReLU activation is followed by a dropout layer layer with a dropout rate of 50%.

The overview of the architecture is displayed in Figure 1. The input words of a single sentence are first tokenized, passed through a pretrained mT5-large encoder (Conneau et al., 2020), and each input word is represented by the embedding of its first subword. Then, the candidate for empty nodes are generated, two per word. The first candidate is generated by passing the input word representations through a 2k-unit dense layer with ReLU activation, a dropout layer, and a 768-unit dense layer. The second candidate is generated by concatenating the first candidate representation with the input word representation and passing the result through an analogous dense-dropout-dense module. Then, three heads are attached, each first passing its input by a ReLU-activated 2k-unit dense layer and dropout: (1) a classification layer deciding whether a candidate actually generates an empty node, (2) a self-attention layer choosing the word order position (i.e., an input word to follow) for every candidate, and (3) a dependency relation classification layer, which processes the candidate representation concatenated with the representation of the word most likely according to the word-order prediction head. Please refer to the released source code for further details.

We train a single multilingual model using the AdaFactor optimizer (Shazeer and Stern, 2018) for 20 epochs, each epoch consisting of 5 000 batches containing 64 sentences each. The learning rate first linearly increases from zero to the peak learning rate of 1e-5 in the first epoch, and then decays to zero in the rest of the training according to a cosine schedule (Loshchilov and Hutter, 2017). Each sentence is sampled from the combination of all corpora containing empty nodes (see Table 1), proportionally to the square root of the word size of the corresponding corpus. The model is trained for 19 hours using a single L40 GPU with 48GB RAM.

The source code is released under the MPL license at https://github.com/ufal/crac2024_zero_nodes_baseline, together with the complete set of used hyperparameters. Furthermore, the trained model is available under the CC BY-SA-NC license at https://www.kaggle.com/models/ufal-mff/crac2024_zero_nodes_baseline/. Finally, the development sets and the test sets of the CorefUD 1.2 datasets with predicted empty nodes are available to all participants of the CRAC 2024 Shared Task.

The intrinsic performance of the baseline system on the development sets of CorefUD 1.2 is presented in Table 1. A predicted empty node is considered correct if it has correct dependency head, dependency relation, and also the word order.

3.2 Coreference Resolution

With the empty nodes predicted by the baseline, we can directly employ the CorPipe 23 from the last year of the shared task (Straka, 2023). The overview of the architecture is presented in Figure 2 and briefly described; for more details, please refer to the original paper.

Treebank	Precison	Recall	F_1-score
ca	92.32	91.01	91.66
cs_pcedt	78.22	59.84	67.81
cs_pdt	81.47	71.56	76.19
cu	81.61	78.76	80.16
es	92.04	91.92	91.98
grc	90.29	86.58	88.39
hu_korkor	74.68	60.21	66.67
hu_szegedkoref	91.93	89.52	90.71
pl	87.50	91.61	89.51
tr	79.05	93.81	85.80

Table 1: Empty nodes prediction baseline performance on the development sets of CorefUD 1.2 corpora containing empty nodes. An empty node is evaluated as correct if it has the correct dependency head, dependency relation, and word order.

CorPipe processes the document one sentence at a time; to provide as much context as possible, as many preceding and at most 50 following tokens are additionally added on input, to the limit of the maximum segment size (512 or 2 560). The words are first passed through a pretrained language encoder model. Then, coreference mentions are predicted using an extension of BIO encoding capable of representing possibly overlapping set of spans. Finally, each predicted mention is represented as a concatenation of its first and last word, and the most likely entity link (possibly to itself) of every mention is generated using a self-attention layer.

During training, the maximum segment size is always 512; however, during inference, we consider also larger segment size of 2 560 for the mT5 models, which support larger segment sizes due to their relative positional embeddings.

3.3 Training

We train the coreference resolution system analogously to the CorPipe 23 training procedure (Straka, 2023). Three model variants are trained, based on either mT5-large, mT5-xl (Xue et al., 2021), or InfoXLM-large (Chi et al., 2021). For every variant, 7 multilingual models are trained on a combination of all corpora, differing only in random initialization. The sentences are sampled proportionally to the square root of the word size of the corresponding corpora.

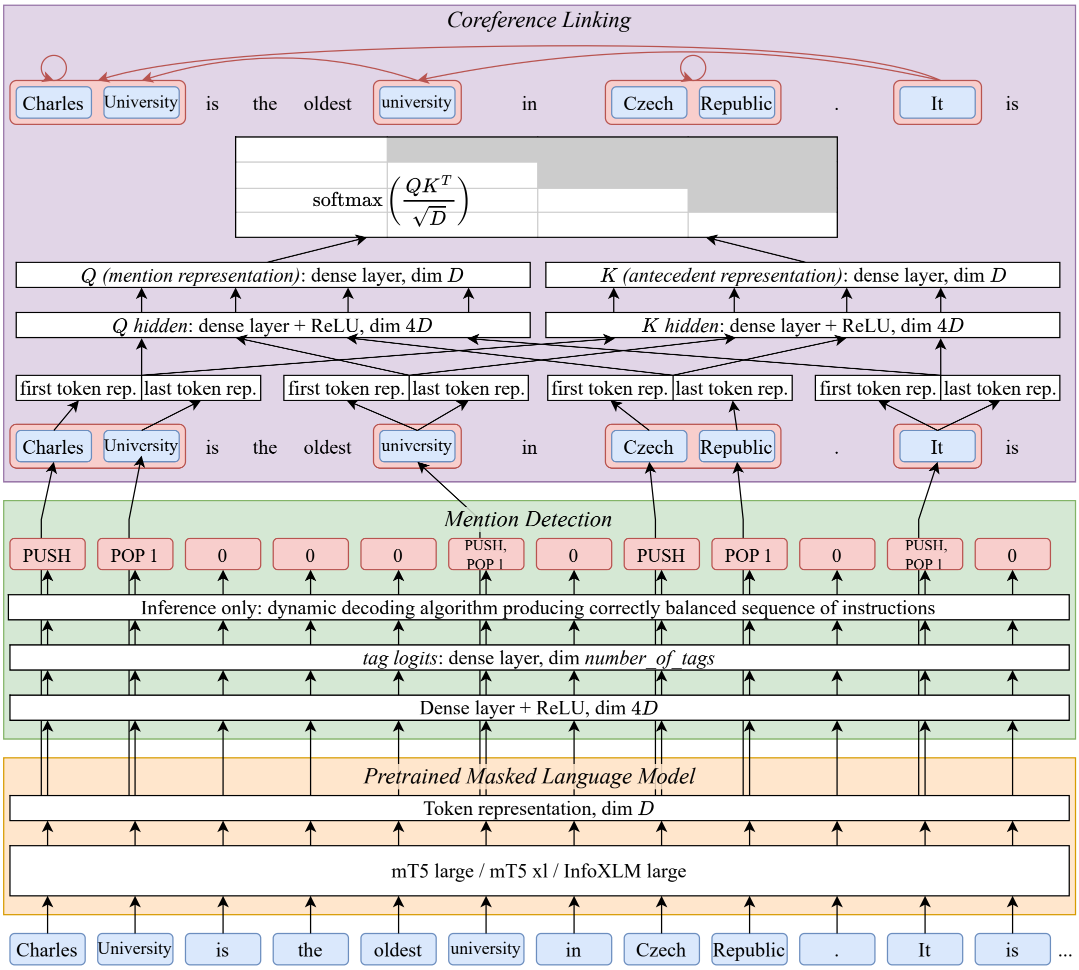
Coreference Linking
Charles
University
is
the
oldest
university
in
Czech
Republic
.
It
is
softmax $\left(\frac{QK^T}{\sqrt{D}}\right)$
Q (mention representation): dense layer, dim D
K (antecedent representation): dense layer, dim D
Q hidden: dense layer + ReLU, dim 4D
K hidden: dense layer + ReLU, dim 4D
first token rep.
last token rep.
Mention Detection
PUSH
POP 1
0
PUSH, POP 1
Inference only: dynamic decoding algorithm producing correctly balanced sequence of instructions
tag logits: dense layer, dim number_of_tags
Dense layer + ReLU, dim 4D
Pretrained Masked Language Model
Token representation, dim D
mT5 large / mT5 xl / InfoXLM large
...

Figure 2: The CorPipe 23 model architecture introduced in Straka (2023).

Every model is trained for 15 epochs, each epoch consisting of 10k batches. The mT5-large and InfoXLM-large variants use the batch size of 8 and train for 14 hours on a single A100 with 40GB RAM; the mT5-xl variant employ the batch size of 12 and train for 17 hours on 4 A100s with 40GB RAM each. The mT5 variants are trained using the AdaFactor optimizer (Shazeer and Stern, 2018) and the InfoXLM-large is trained using Adam (Kingma and Ba, 2015). The learning rate is first increased from 0 to the peak learning rate in the first 10% of the training and then decayed according to the cosine schedule (Loshchilov and Hutter, 2017); we employ the peak learning rates of 6e-4, 5e-4, and 2e-5 for the mT5-large, mT5-xl, and InfoXLM-large encoders, respectively.

For each model, we keep the checkpoints after every epoch, obtaining a pool of $3 \cdot 7 \cdot 15$ checkpoints. From this pool, we select three configurations: (1) a single checkpoint reaching the highest development score on all the corpora, (2) a best-performing checkpoint for every corpus according to its development set, (3) an ensemble of 5 best-performing checkpoints for every corpus.

4 Single-stage CorPipe

While the two-stage variant is full-fledged, allowing coreference mention to be composed of any continual sequence of input words and empty nodes, it requires two large pretrained encoders, which makes the model about twice as big and twice as slow compared to a single model.

Therefore, we also propose a single-stage variant, with the goal of using just a single pretrained language encoder model. For simplicity's sake, we restrict the model in the following way: if a coreference mention contains an empty node, the whole mention must be just this single empty node. In other words, a coreference mention either does not contain empty nodes, or it is just a single empty node. Note that this restriction does not decrease the score under the head-match metric because only the mention head is used during score computation.

With the described restriction, we no longer need

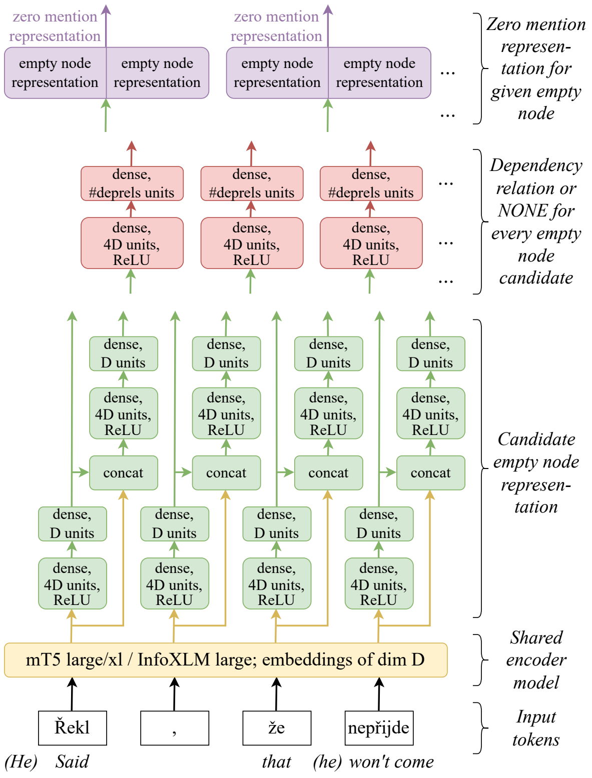
zero mention representation
empty node representation
empty node representation
zero mention representation
empty node representation
empty node representation
...
...
dense, #deprels units
dense, #deprels units
dense, #deprels units
...
dense, 4D units, ReLU
dense, 4D units, ReLU
dense, 4D units, ReLU
...
...
dense, D units
dense, 4D units, ReLU
concat
dense, D units
dense, 4D units, ReLU
mT5 large/xl / InfoXLM large; embeddings of dim D
Řekl
,
že
nepřijde
(He) Said
that
(he) won't come
Zero mention represen-tation for given empty node
Dependency relation or NONE for every empty node candidate
Candidate empty node represen-tation
Shared encoder model
Input tokens

Figure 3: The changes in the CorPipe 23 architecture when empty nodes and zero mentions are generated jointly with mention detection and coreference linking.

to distinguish between empty nodes and zero coreference mentions; therefore, the single-stage model predicts only such empty nodes that are also zero coreference mentions. Finally, the word order of an empty node is no longer needed for evaluation; as a result, we no longer predict the word order explicitly and place the empty node after its dependency head in the word order.

In Figure 3, we visualize the proposed changes to the CorPipe architecture needed to support joint empty nodes/zero mentions prediction. Analogously to the empty nodes baseline described in Section 3.1, we start by generating two candidate empty nodes representations from every input word representation. We then run a classification head for every candidate, which either predicts NONE when the candidate should not generate an empty node, or it predicts the dependency relation of the generated empty node. Finally, to construct a representation of a zero coreference mention, we concatenate the empty node representation to itself because the empty node is both the first and the last word of the mention. The coreference linking then proceeds as before, just using a concatenation of surface mentions and zero mentions.

The single-stage model is trained analogously to the two-stage model. The only differences are that (1) we pass only the input words through the pretrained language encoder model, (2) we add the loss of the classifier predicting dependency relation or NONE to the other losses (using simple addition), and (3) we concatenate the zero mention representations to the surface mention representations before the coreference linking step.

We closely follow the training procedure of the two-stage model described in Section 3.3. Notably, we also consider the same three pretrained encoders, train the same number of models using the same optimizers and learning rates, and select the same three configurations (single best-performing checkpoint, per-corpus best checkpoint, and a per-corpus 3-model ensemble).[2]

5 Shared Task Results

In the shared task, each team was allowed to submit at most three systems. We submitted the following configurations:

- **CorPipe-single**, the large-sized single-stage model checkpoint achieving the best development performance across all corpora;
- **CorPipe**, the best-performing 3-model single-stage ensemble for every corpus;
- **CorPipe-2stage**, the best-performing 5-model two-stage ensemble for every corpus.

The first configuration corresponds to a real-world deployment scenario, where a single model would be used for all corpora; the latter configurations are the highest performing single-stage approach (**CorPipe**, Section 4) and two-stage approach (**CorPipe-2stage**, Section 3).

The official results of the shared task's primary metric are presented in Table 2. All our submissions outperform other participant systems, even if **CorPipe-single** only slightly. Overall, the ensembled single-stage variant outperforms other participants by 2.8 percent points, and the ensembled two-stage variant outperforms other participants by 3.9 percent points.

Table 3 shows the results of the submitted systems using four metrics. Apart from the primary head-match metric, our three submissions outperform all others also when evaluated using exact match and with singletons. When considering par-

[2] We only managed to use a 3-model ensemble before the shared task deadline, while we use a 5-model ensemble for the two-stage variant.

System	Avg	ca	cs pced	cs pdt	cu	de parc	de pots	en gum	en litb	en parc	es	fr	grc	hbo	hu kork	hu szeg	lt	no bokm	no nyno	pl	ru	tr
CorPipe-2stage	**73.90** **1**	82.2 2	**74.8** **1**	**77.2** **1**	**61.6** **1**	69.5 3	71.8 2	**75.7** **1**	**79.6** **1**	68.9 2	**82.5** **1**	68.2 2	**71.3** **1**	**72.0** **1**	63.2 2	**70.0** **1**	**75.8** **1**	**79.8** **1**	**78.0** **1**	**78.5** **1**	**83.2** **1**	**68.2** **1**
CorPipe	72.75 2	81.0 3	73.7 2	75.8 2	60.7 2	**71.7** **1**	71.5 3	74.6 2	79.1 2	**69.8** **1**	81.0 3	**68.8** **1**	68.5 2	70.9 2	60.3 3	68.1 3	75.8 2	79.5 2	77.5 2	77.0 2	83.1 2	59.4 3
CorPipe-single	70.18 3	80.4 4	72.8 3	74.8 4	57.1 3	61.6 4	67.0 4	74.4 3	78.1 3	58.6 3	79.8 4	67.9 3	66.0 3	67.2 3	60.1 4	67.3 4	75.2 3	78.9 3	76.6 3	75.2 4	81.2 3	53.4 4
Ondfa	69.97 4	**82.5** **1**	70.8 4	75.8 3	55.0 4	71.4 2	**71.9** **1**	70.5 4	74.2 4	55.6 4	81.9 2	62.7 4	61.6 4	61.6 4	**64.9** **1**	69.3 2	72.0 4	74.5 4	72.1 4	76.3 3	80.5 4	64.5 2
BASELINE†	53.16 5	68.3 5	64.1 5	63.8 5	24.5 5	47.2 5	55.6 5	63.2 5	63.5 5	33.1 6	69.6 5	53.6 5	28.8 5	24.6 6	35.1 5	54.5 5	62.0 5	65.0 5	63.7 5	66.2 5	65.8 5	44.0 5
DFKI-CorefGen	33.38 6	34.8 6	32.9 6	30.9 6	22.5 6	23.1 7	45.9 7	35.5 6	46.6 6	32.7 7	37.8 6	36.3 7	25.9 6	38.0 5	23.5 7	33.9 6	42.7 7	37.9 6	35.7 6	27.2 6	47.8 7	9.7 6
Ritwikmishra	16.47 7	0.0 7	0.0 7	0.0 7	6.8 7	25.4 6	48.9 6	0.0 7	0.0 7	53.1 5	0.0 7	43.7 6	5.6 7	0.1 7	33.4 6	30.3 7	44.8 6	0.0 7	0.0 7	0.0 7	53.9 6	0.0 7

Table 2: Official results of CRAC 2024 Shared Task on the test set (CoNLL score in %). The system † is described in Pražák et al. (2021); the rest in Novák et al. (2024).

System	Head-match	Partial-match	Exact-match	With Sin-gletons
CorPipe-2stage	**73.90** **1**	**72.19** **1**	**69.86** **1**	**75.65** **1**
CorPipe	72.75 2	70.30 2	68.36 2	74.65 2
CorPipe-single	70.18 3	68.02 4	66.07 3	71.96 3
Ondfa	69.97 4	69.82 3	40.25 5	70.67 4
BASELINE	53.16 5	52.48 5	51.26 4	46.45 5
DFKI-CorefGen	33.38 6	32.36 6	30.71 6	38.65 6
Ritwikmishra	16.47 7	16.65 7	14.16 7	15.42 7

Table 3: Official results of CRAC 2024 Shared Task on the test set with various metrics in %.

tial match, the CorPipe-single is outperformed by the system Ondfa, assumingly because it limits the predicted mentions just to their heads, which slightly improves partial match but severely deteriorates exact match.

6 Ablations Experiments

6.1 CorefUD 1.2

Table 4 contains quantitative analysis of ablation experiments on the CorefUD 1.2 test set. In Table 4.A, we compare the three configurations of the single-stage model variant. Selecting the best-performing checkpoint for every corpus increases the overall score by 1.4 percent points, while making the model up to 21 times larger. Further addition of ensembling improves the score by another 1.2 percent points.

The same comparison is available also for the two-stage model variant in Table 4.B. We observe a similar trend of 1.2 percent points increase for the best per-corpus checkpoint approach and further 1.4 percent points increase during ensembling.

The sections C, D, and E of Table 4 compare the individual checkpoint configurations of the single-stage and the two-stage models. We observe that the effect of the two-stage model is 0.9–1.1 percent point increase in all checkpoint configuration. We hypothesize that two factors contribute to the better performance of the two-stage variant: first, the empty node representation is computed by a pretrained encoder, allowing better contextualization of the empty node representation. Second, the mentions with empty nodes are represented in the original form, i.e., the mentions can contain any sequence of input words and empty nodes, while the single-stage variant represent zero mentions always by a single empty node.

It would be interesting to evaluate the two-stage variant using the gold empty nodes instead of predicted empty nodes to quantify the decrease of the score caused by empty node prediction errors. Unfortunately, such an evaluation is not supported by the shared task evaluation platform. Nevertheless, Table 4.F at least shows that such a difference for the provided baseline coreference system (Pražák et al., 2021) is 1.4 percent points, as reported by the shared task organizers.

Finally, meaningful comparison of the shared task results between this year and the last year is very difficult to carry out. While many corpora have changed only marginally and the evaluation metric is the same (so the results are reasonably comparable), other corpora have changed substantially (especially Polish and Turkish). Even so, we provide numerical comparison of this year's and last year's best systems in Table 4.G. This year's results are slightly worse than in the last year, on

System	Avg	ca	cs pced	cs pdt	cu	de parc	de pots	en gum	en litb	en parc	es	fr	grc	hbo	hu kork	hu szeg	lt	no bokm	no nyno	pl	ru	tr
A) CorPipe Single-stage Variants																						
Single model	70.18	80.4	72.8	74.8	57.1	61.6	67.0	74.4	78.1	58.6	79.8	67.9	66.0	67.2	60.1	67.3	75.2	78.9	76.6	75.2	81.2	53.4
Per-corpus best	+1.42	–0.4	–0.6	–0.2	+2.5	+7.2	+2.7	–0.4	–0.6	+10.4	–0.0	–0.3	+1.0	+1.5	**+2.5**	–1.6	**+0.9**	–0.4	+0.9	–0.2	–0.2	+5.1
Per-corpus ensemble	**+2.62**	**+0.6**	**+0.9**	**+1.0**	**+3.6**	**+10.1**	**+4.5**	**+0.2**	**+1.0**	**+11.2**	**+1.2**	**+0.9**	**+2.5**	**+3.7**	+0.2	**+0.8**	+0.6	**+0.6**	**+0.9**	**+1.8**	**+1.9**	**+6.0**
B) CorPipe Two-stage Variants																						
Single model	71.32	81.0	74.2	75.9	56.7	64.7	66.4	74.7	78.2	57.9	81.2	67.2	67.6	64.2	61.6	67.9	**77.7**	77.6	77.3	77.4	81.3	67.0
Per-corpus best	+1.18	+0.1	+0.4	+0.3	+3.7	**+4.9**	+0.6	–1.2	+0.5	+10.2	+0.7	–0.2	+1.3	+5.6	–0.2	–0.6	–4.2	+2.2	+0.4	+0.5	–0.1	+0.2
Per-corpus ensemble	**+2.58**	**+1.2**	**+0.6**	**+1.3**	**+4.9**	+4.8	**+5.4**	**+1.0**	**+1.4**	**+11.1**	**+1.3**	**+1.0**	**+3.7**	**+7.8**	**+1.6**	**+2.1**	–1.9	**+2.2**	**+0.7**	**+1.1**	**+1.9**	**+1.2**
C) Comparison of Single-model Variants																						
Single-stage	70.18	80.4	72.8	74.8	**57.1**	61.6	**67.0**	74.4	78.1	**58.6**	79.8	**67.9**	66.0	**67.2**	60.1	67.3	75.2	**78.9**	76.6	75.2	81.2	53.4
Two-stage	**+1.12**	**+0.6**	**+1.4**	**+1.1**	–0.4	**+3.1**	–0.6	**+0.3**	**+0.1**	–0.7	**+1.5**	–0.7	**+1.6**	–3.0	**+1.5**	**+0.6**	**+2.5**	–1.3	**+0.7**	**+2.2**	**+0.1**	**+13.6**
D) Comparison of Per-corpus Best Variants																						
Single-stage	71.59	80.0	72.2	74.6	59.6	68.8	**69.7**	**74.0**	77.5	**69.0**	79.7	**67.6**	67.0	68.7	**62.6**	65.7	**76.1**	78.5	77.5	75.0	81.0	58.5
Two-stage	**+0.91**	**+1.1**	**+2.4**	**+1.6**	**+0.8**	**+0.8**	–2.7	–0.5	**+1.2**	–0.9	**+2.2**	–0.6	**+1.9**	**+1.1**	–1.2	**+1.6**	–2.6	**+1.3**	**+0.2**	**+2.9**	**+0.2**	**+8.8**
E) Comparison of Per-corpus Ensemble Variants																						
Single-stage	72.75	81.0	73.7	75.8	60.7	**71.7**	71.5	74.6	79.1	**69.8**	81.0	**68.8**	68.5	70.9	60.3	68.1	75.8	79.5	77.5	77.0	83.1	59.4
Two-stage	**+1.15**	**+1.2**	**+1.1**	**+1.4**	**+0.9**	–2.2	**+0.3**	**+1.1**	**+0.5**	–0.8	**+1.5**	–0.6	**+2.8**	**+1.1**	**+2.9**	**+1.9**	**+0.0**	**+0.2**	**+0.5**	**+1.5**	**+0.1**	**+8.8**
F) Comparison of the Baseline System with Gold and Predicted Empty Nodes																						
Predicted empty nodes	53.16	68.3	64.1	63.8	24.5	**47.2**	**55.6**	**63.2**	**63.5**	**33.1**	69.6	**53.6**	28.8	**24.6**	35.1	54.5	**62.0**	**65.0**	**63.7**	66.2	**65.8**	44.0
Gold empty nodes	**+1.44**	**+1.3**	**+4.8**	**+2.4**	**+3.1**	0.0	0.0	0.0	0.0	0.0	**+1.0**	0.0	**+3.1**	0.0	**+6.5**	**+0.1**	0.0	0.0	0.0	**+0.8**	0.0	**+7.2**
G) Comparison of the CorPipe-2stage Ensemble System and the CRAC23 Best Results																						
CorPipe-2stage, ensemble	74.55	82.2	74.8	77.2	—	69.5	71.8	75.7	—	68.9	82.5	68.2	—	—	63.2	70.0	75.8	**79.8**	78.0	78.5	**83.2**	**68.2**
CorPipe23, CRAC23	**+0.65**	**+1.0**	**+4.5**	**+2.3**	—	**+1.5**	**+0.0**	**+0.8**	—	**+2.1**	**+1.0**	**+0.4**	—	—	**+6.3**	**+0.8**	**+0.6**	–0.2	**+1.0**	**+1.3**	–0.6	–11.7

Table 4: Ablations experiments on the CorefUD 1.2 test set (CoNLL score in %).

Paper	Model	#model	∅, ELMO,	large PLM	xl PLM	xxl PLM
		calls	base PLM	∼350M	∼3B	∼11B
(Lee et al., 2017)	e2e	1	67.2$_{\varnothing}$			
(Lee et al., 2018)	e2e	1	70.4$_{\text{ELMO}}$			
(Lee et al., 2018)	c2f	1	73.0$_{\text{ELMO}}$			
(Joshi et al., 2019)	c2f	1	73.9$_{\text{BERT}}$	76.9$_{\text{BERT}}$		
(Joshi et al., 2020)	c2f	1		79.6$_{\text{SpanBERT}}$		
(Kirstain et al., 2021)	s2e	1		80.3$_{\text{Longformer}}$		
(Otmazgin et al., 2023)	s2e/LingMess	1		81.4$^{\text{+additional annotations}}_{\text{Longformer}}$		
(Dobrovolskii, 2021)	WL	1		81.0$_{\text{RoBERTa}}$		
(D'Oosterlinck et al., 2023)	WL/CAW	1		81.6$_{\text{RoBERTa}}$		
(Liu et al., 2022)	ASP	$\mathcal{O}(n)$	76.6$_{\text{T5}}$	79.3$_{\text{T5}}$	82.3$_{\text{T0}}$	82.5$_{\text{FlanT5}}$
(Bohnet et al., 2023)	seq2seq	$\mathcal{O}(n)$			78.0$^{\text{dev}}_{\text{mT5}}$	83.3$_{\text{mT5}}$
(Wu et al., 2020)	CorefQA	$\mathcal{O}(n)$	79.9$^{\text{+QA data}}_{\text{SpanBERT}}$	83.1$^{\text{+QA data}}_{\text{SpanBERT}}$		
This paper	CorPipe	1		80.7$_{\text{T5}}$	82.0$_{\text{FlanT5}}$	
This paper	CorPipe	1		77.2$_{\text{mT5}}$	78.9$_{\text{mT5}}$	

Table 5: Comparison of CorPipe and other models on OntoNotes, using pretrained models of various size.

average by 0.65 percent points, but the difference is quite comparable to the effect of predicted/gold empty nodes on the baseline system (cf. Table 4.F).

6.2 OntoNotes

To compare the performance of the CorPipe architecture to English state-of-the-art models, we train also models on the OntoNotes dataset (Pradhan et al., 2013). The dataset does not contain any empty nodes, so we use the last year's training setup, with the two exceptions: we also consider pretrained English-specific encoders T5 (Raffel et al., 2020) and Flan-T5 (Chung et al., 2024), and we consider larger segment size during training (up to 1 536 subwords).

The results are presented in Table 5. In the large-sized setting, CorPipe outperforms all models except models utilizing additional data (Otmazgin et al., 2023; Wu et al., 2020) and models utilizing the word-level approach (Dobrovolskii, 2021; D'Oosterlinck et al., 2023).[3] In the xl-sized settings, our model is 0.3 percent points below the state of the art of Liu et al. (2022); notably, CorPipe outperforms the state of the art system Bohnet et al. (2023) and all large-sized models not using additional training data. Unfortunately, we did not have the resources to train an xxl-sized model.

7 Conclusions

We presented CorPipe 24, the winning entry to the CRAC 2024 Shared Task on Multilingual Coreference Resolution (Novák et al., 2024). Our system has two variants, either first predicting empty nodes using a pretrained language encoder model and then performing coreference resolution employing another pretrained model, or predicting the empty nodes jointly with mention detection and coreference linking. Both variants surpass other participants by a large margin of 3.9 and 2.8 percent points, respectively. The source code and the trained model are available at `https://github.com/ufal/crac2024-corpipe`.

Acknowledgements

This work has been supported by the Grant Agency of the Czech Republic, project EXPRO LUSyD (GX20-16819X), and has been using data provided by the LINDAT/CLARIAH-CZ Research Infrastructure (`https://lindat.cz`) of the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2023062).

Limitations

The presented system has demonstrated its performance only on a limited set of 15 languages, and heavily depends on a large pretrained model, transitively receiving its limitations and biases.

Training with the mT5-large pretrained model requires a 40GB GPU, which we consider affordable; however, training with the mT5-xl pretrained model needs nearly four times as much GPU memory.

[3] We are of course curious to find out how the word-level approach works on the CorefUD dataset. Nevertheless, we hypothesize that on some of the CorefUD corpora it might not work well because the mention heads in these corpora are considerably less unique than in OntoNotes.

References

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.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Vladimir Dobrovolskii. 2021. Word-level coreference resolution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7670–7675, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Karel D'Oosterlinck, Semere Kiros Bitew, Brandon Papineau, Christopher Potts, Thomas Demeester, and Chris Develder. 2023. CAW-coref: 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, Singapore. Association for Computational Linguistics.

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

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

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

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

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

Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Michal Novák, Barbora Dohnalová, Miloslav Konopík, Anna Nedoluzhko, Martin Popel, Ondřej Pražák, 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 (CRAC 2024)*, Miami, Florida, USA. Association for Computational Linguistics.

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

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

Ondřej Pražák and Miloslav Konopik. 2022. End-to-end multilingual coreference resolution with mention head prediction. In *Proceedings of the CRAC 2022 Shared Task on Multilingual Coreference Resolution*, pages 23–27, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Ondřej Pražák, Miloslav Konopík, and Jakub Sido. 2021. Multilingual coreference resolution with harmonized annotations. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1119–1123, Held Online. INCOMA Ltd.

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.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive Learning Rates with Sublinear Memory Cost. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4603–4611. PMLR.

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.

Milan Straka and Jana Straková. 2022. ÚFAL CorPipe at CRAC 2022: Effectivity of multilingual models for coreference resolution. In *Proceedings of the CRAC 2022 Shared Task on Multilingual Coreference Resolution*, pages 28–37, Gyeongju, Republic of Korea. Association for Computational Linguistics.

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

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

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, Singapore. Association for Computational Linguistics.

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, Gyeongju, Republic of Korea. Association for Computational Linguistics.