End-to-end Multilingual Coreference Resolution with Headword Mention Representation

Ondřej Pražák and **Miloslav Konopík**

{ondfa,konopik}@kiv.zcu.cz

Department of Computer Science and Engineering,
NTIS – New Technologies for the Information Society,
Faculty of Applied Sciences, University of West Bohemia, Technická 8, 306 14 Plzeň
Czech Republic

Abstract

This paper describes our approach to the CRAC 2024 Shared Task on Multilingual Coreference Resolution. Our model is based on an end-to-end coreference resolution system. Apart from joined multilingual training, we improved our results with headword mention representation and training large model mT5-xxl through LORA. We provide an analysis of the performance of our model. Our system ended up in 4^{th} place. Moreover, we reached the best performance on three datasets out of 21.

1 Introduction

Coreference resolution is the task of finding language expressions that refer to the same real-world entity (antecedent) within a given text. These coreferential expressions can either originate from a single sentence or be separated by one or more sentences. In some challenging cases, it is necessary to consider the entire document to determine whether two expressions refer to the same entity accurately. This task can be divided into two subtasks. Identify entity mentions and group them together according to the real-world entity they refer to. The task of coreference resolution is closely related to anaphora resolution – see (Sukthanker et al., 2020) to compare these two tasks.

This paper describes our approach to the CRAC 2024 Shared Task on Multilingual Coreference Resolution (Novák et al., 2024), which is the third edition of this shared task. The task is based on the CorefUD dataset (Nedoluzhko et al., 2022). The CorefUD corpus, currently at version 1.2, comprises 21 different datasets across 15 languages in a harmonized scheme. Table 1 shows basic statistics of the corpus As CorefUD is meant to be the extension of Universal Dependencies for coreference annotation, all the datasets in CorefUD are treebanks. In the current version of the dataset, all dependency relations were obtained from an automatic parser. The coreference annotation is built upon the dependencies. This means that the mentions are subtrees in the dependency tree and can be represented with the head. In fact, in some of the datasets, there are non-treelet mentions – those that do not form a single subtree. But even for these non-treelet mentions, a single headword is selected. Non-tree mentions arise because some datasets were not annotated in a treebank form - the annotators were asked to find mentions as continuous spans, and the syntactic information was added during the harmonization. Notable differences exist among the datasets. One of the most prominent ones is the presence of singletons. Singletons are clusters that contain only one mention; therefore, they are not part of any coreference relation, yet they are annotated as mentions. Please see Nedoluzhko et al. (2022) or Nedoluzhko et al. (2021) for details about the dataset. The task was simplified to predict only non-singleton mentions and group them into entity clusters.

For evaluation, the CorefUD scorer[1] is provided. The primary evaluation score is the CoNLL F_1 score with head matching and singletons excluded. In the CorefUD scorer, a system mention matches a gold mention only if they share the same headword.

Participants should also predict the empty nodes for zero mentions this year. In previous years (Žabokrtský et al., 2022; Žabokrtský et al., 2023), gold empty nodes were provided. However, the organizers provide a baseline for predicting empty nodes. Due to time limitations, we focused just on coreference resolution, and we used empty nodes predicted by a baseline system.

2 Related Work

Since many of the datasets in the CorefUD collection do not contain singletons annotation, we believe that the end-to-end approach is the best

[1] https://github.com/ufal/corefud-scorer

Proceedings of the Seventh Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2024), pages 107–113
November 15, 2024. ©2024 Association for Computational Linguistics

CorefUD dataset	Total size				
	docs	sents	words	empty	singletons
Ancient_Greek-PROIEL	19	6,475	64,111	6,283	0,0%
Ancient_Hebrew-PTNK	40	1,161	28,485	0	57.9%
Catalan-AnCora	1550	16,678	488,379	6,377	74.6%
Czech-PDT	3165	49,428	834,721	33,086	35.3%
Czech-PCEDT	2312	49,208	1,155,755	45,158	1.4%
English-GUM	150	7,408	134,474	0	75%
English-LitBank	100	8,560	210,530	0	72.8%
English-ParCorFull	19	543	10,798	0	6.1%
French-Democrat	126	13,054	284,823	0	81.8%
German-ParCorFull	19	543	10,602	0	5.8%
German-PotsdamCC	176	2,238	33,222	0	76.5%
Hungarian-KorKor	94	1,351	24,568	1,988	0.9%
Hungarian-SzegedKoref	400	8,820	123,976	4,849	7.9%
Lithuanian-LCC	100	1,714	37,014	0	11.2%
Norwegian-BokmaalNARC	346	15,742	245,515	0	89.4%
Norwegian-NynorskNARC	394	12,481	206,660	0	88.7%
Old_Church_Slavonic-PROIEL	26	6,832	61,759	6,289	0,0%
Polish-PCC	1828	35,874	538,891	864	82.6%
Russian-RuCor	181	9,035	156,636	0	2.5%
Spanish-AnCora	1635	17,662	517,258	8,111	73.4%
Turkish-ITCC	24	4,733	55,341	0	1.0%

Table 1: Dataset Statistics

choice. On the other hand, the best system in the previous year (Straka, 2023) is a two-stage model using extended BIO schema for mention identification.

Most of the end-to-end approaches are built upon Lee et al. (2017) who originally proposed to go over all possible spans and classify coreferences directly on these spans. As our model is also based on this, we will describe more details later. Many modifications of this model have been proposed mainly focusing on better text encoding (span representation), model optimization and higher-order model (Lee et al., 2018; Joshi et al., 2019; Xu and Choi, 2020; Joshi et al., 2020).

Dobrovolskii (2021) proposed to reduce mention space be selecting a single word to represent each mention. They use the syntactic head as mention representative. They perform experiments on the English OntoNotes corpus. To reconstruct the original mentions, they use a CNN-based span predictor in a subsequent step after antecedent prediction.

Hu et al. (2022) proposed low-rank adaptation as one of the most common techniques for efficient fine-tuning by reducing the number of trainable parameters with factorization. The original idea to use this in Transformer fine-tuning comes from Adapters (Houlsby et al., 2019). The schema of LORA is shown in Figure 1. We reduce the number of trainable parameters by freezing the original model and adding a small layer between all fully connected layers. The first weight matrix is initialized randomly, and the second is set to zero to preserve the original output at the initial step. By reducing the number of trainable parameters, LoRA reduces memory requirements and prevents overfitting but preserves a lot of original computational capability since weights on every layer can be changed during finetuning.

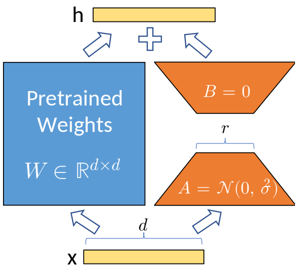
h
Pretrained Weights
$W \in \mathbb{R}^{d\times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

Figure 1: LoRA schema, taken from Hu et al. (2022)

Model	Pretrained params	New params
mBERT	180M	15M
XLM-R	555M	20M
mT5	5.7B	70-400M

Table 2: Number of trainable parameters of the models

3 Model

Our model builds on the official transformer-based end-to-end baseline (Pražák et al., 2021). It is based on the CRAC 2022 participating system (Pražák and Konopik, 2022) and its extension (Pražák and Konopík, 2024) with all the proposed modifications. The underlying neural end-to-end coreference resolution model was originally proposed by Lee et al. (2017). The model predicts the antecedents directly from all possible mention spans without a previous discrete decision about mentions. In the training phase, it maximizes the marginal log-likelihood of all correct antecedents:

$$J(D) = \log \prod_{i=1}^{N} \sum_{\hat{y} \in Y(i) \cap \texttt{GOLD}(i)} P(\hat{y}) \qquad (1)$$

where GOLD(i) is the set of spans in the training data that are antecedents.

The model performs well on the OntoNotes dataset, where singletons are not annotated. We believe the model is optimal for the CorefUD dataset as well since some of the CorefUD datasets do not contain singletons. Moreover, the primary evaluation metric ignores singletons, so it does not matter that the model is not able to predict them. However, employing singletons in the model can improve mention identification capabilities of the model on the datasets where some singletons are annotated.

Here, we just describe the most significant extensions of the basic model. For a detailed description of all the extensions together with a deep analysis of their benefits, please refer to Pražák and Konopík (2024).

Employed Models We based our model on two encoders of different sizes; XLM Roberta large (Conneau et al., 2020), and mT5-xxl (Xue et al., 2021) (only the encoder part). Both models are significantly larger than the original BERT (Devlin et al., 2018) The number of parameters is provided in Table 2.

Joined Model Pretraining As you can see from Table 2, approximately 17 million parameters are trained from scratch for XLM-R and 70M for mT5 (the upper bound 400M is including adapter weights which are technically trained from scratch, but the original pretrained parameters makes them much easier to train). For smaller datasets, training so many random parameters is practically impossible. To solve this issue, we first pre-train the model on the joined dataset and then fine-tune the model for a specific language.

Heads Mention Representation As mentioned above, the official scorer uses head-match evaluation. Inspired by word-level coreference resolution (Dobrovolskii, 2021), we decided to use only headwords for mention representation. Since the mentions are considered the same if they have the same head, we do not need the span reconstruction step as in Dobrovolskii (2021). As pointed out by (Dobrovolskii, 2021), a single-word representation reduces the mention space from quadratic to linear, and the model is learning more effectively. There are also much fewer 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. The syntactic information should be even more beneficial in case of heads mention representation. for the model, so we use it for all the datasets.

The whole model stays practically the same, we just change the span extraction step where we consider all words in the document as potential mentions.

Singletons Some datasets in the CorefUD collection have singletons annotated, and others do not. Specifically, in CorefUD 1.2, 10 out of 21 datasets have more than 10% singletons, and 8 of these have more than 70% singletons, which is probably a sign of consistent entity annotation independent of the coreference annotation. The original model by Lee et al. (2017) completely ignores singletons during training[2]. As a result, for these eight singleton-including datasets, we discard more than 70% of training data for mention identification task. To leverage this data, Pražák and Konopík (2024) incorporated singleton modeling into the model. They modify the loss function to model mentions

[2] The loss is the sum over all correct antecedents, and since singletons have no gold antecedents, they do not affect the loss

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 as formalized in Equation 2. 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.

Large Model For Training the large model (mT5-xxl) we suggest using LORA. We propose two variants. In the first we use LORA for both joined pretraining and fine-tuning on individual datasets. In the second variant, we use traditional training of all the parameters in the joined pretraining phase and LORA only for fine-tuning on individual datasets. We tried several different values for LORA rank (size of the adapter layer) from 8 to 128

4 Training

We trained all the models on NVIDIA A40 graphic cards using online learning (batch size 1 document). We limit the maximum sequence length to 8 segments of 512 tokens. During training, if the document is longer than 8×512 tokens, a random segment offset is sampled to take a random continuous block of 8 segments, and the rest of them are discarded. During prediction, longer documents are split into sub-documents overlapping in one segment, which is then used to merge the coreference clusters from all the sub-documents. More details can be found in Pražák and Konopík (2024). We use 80k steps for model pre-training on all the datasets and approximately 30k for fine-tuning on each dataset. Pretraining took 24 hours and fine-tuning 2-6 hours.

5 Results & Discussion

Results of several variants of our model are presented in Table 3.

The table is divided into four sections, the first two comparing the results of different encoders (XLMR Roberta large and mT5-xxl). XLMR column has two variants, one using headwords as mention representations and the other using the whole spans. mT5 column contains two variants described in Section 3, full weights updating from pretraining and LORA even for pretraining. The third section contains results when selecting the best model on dev data. It contains the version *submitted* to the shared task and the version with optimal hyperparameter setting according to Pražák and Konopík (2024). The last section describes the same settings as the third one evaluated on CorefUD 1.1 (from CRAC 2023).

When we compare the first two sections, we can see that XLM-R achieves better results for some datasets than mT5; for others, it is the opposite. Generally, we can say that XLM-R is better for smaller datasets and mT5 for larger ones. This trend would suggest that mT5 is overfitted on smaller datasets. We tried many different values of the LORA factor and all the regularization parameters, but it did not yield better results. The larger model is harder to train, and we might not find the best combination of hyperparameters.

Full joined pretraining of mT5 is better than the LORA variant for all the datasets except for *en-parcorfull*, which we consider an anomaly.

FullSpan is surprisingly better than heads-only representation on *de-parcorfull* dataset. Again, we consider this an anomaly. ParCor datasets are the smallest ones in the collection and results on these datasets are very noisy. On average, *FullSpan* is almost 3% below heads-only. It is actually better for more datasets but this is caused by a mistake. We trained the model in the configuration from Pražák and Konopík (2024), so the model is not directly comparable to *XLMR-heads* column, but it is comparable to *BEST-dev-paper24* column. We did not have enough time to rerun the experiment.

We can compare the results for individual datasets between CorefUD 1.1 and CorefUD 1.2 from the last two sections of the Table. As expected, we can observe a performance drop from 1-4% for all datasets with empty nodes. On the other hand, we can see improvement for some datasets. It is known that there were mistakes in the Turkish dataset, where the improvement is most significant. Another significant improvement is there for Lithuanian.

One more thing worth noticing. Our model is much worse for newly added ancient languages than for the rest of the datasets (compared to *CorPipe*). We believed this was caused by a bug in the submitted version where we forgot to add new languages into joined pretraining. However, after fixing this, the results are very similar. We won-

Dataset/Model	XLMR		mT5		BEST-dev		CRAC23	
	FullSpan	heads	Full	LORA	submited	paper24	Submited	paper24
ca_ancora	75.29	80.58	**82.18**	80.11	82.37	81.29	75.49	82.57
cs_pcedt	68.96	**71.13**	67.67	62.46	71.13	73.5	77.37	78.46
cs_pdt	74.72	**77.14**	74.58	69.19	77.14	77.1	76.67	80.09
cu_proiel	43.23	54.24	44.53	44.61	54.24	53.2		
de_parcorfull	**81.23**	79.44	79.9	77.83	81.61	78.34	80.45	80.25
de_potsdamcc	76.77	76.76	**79.23**	75.08	79.23	77.41	78.17	77.95
en_gum	73.72	74.36	**75.98**	71.31	75.98	75.66	73.67	76
en_litbank	66.44	71.17	**73.31**	68.04	74.47	71.29		
en_parcorfull	**76.89**	70.81	69.84	70.32	70.81	70.51	67.92	67.41
es_ancora	76.81	81.4	**81.94**	79.63	82.08	81.61	77.62	82.92
fr_democrat	66.47	**65.72**	65.41	61.95	66.57	69.31	64.47	70.35
grc_proiel	58.22	**64.54**	60.25	59.18	64.54	63.1		
hbo_ptnk	46.25	59.68	**61.83**	59.8	63.44	56.93		
hu_korkor	65.58	**70.04**	**70.01**	65.22	70.69	69.9	70.55	74.01
hu_szegedkoref	68.03	69.89	69.53	69.2	70.25	70.08	68.82	70.9
lt_lcc	**78.44**	76.68	76.3	74.46	76.3	78.99	76.41	76.91
no_bokmaalnarc	76.64	77.25	**78**	74.77	79.21	78.02	76.48	78.62
no_nynorsknarc	77.88	78.72	78.41	75.06	78.72	78.59	77.55	80.41
pl_pcc	75.04	74.88	**76.07**	73.8	76.25	75.14	75.67	76.16
ru_rucor	74.31	73.45	**74.24**	71.69	75.03	75.96	70.03	77.56
tr_itcc	55	55.07	54.9	47.26	58.35	59.77	43.9	53.72
avg	69.33	71.57	71.15	68.14	72.78	72.18	72.43	75.55

Table 3: Results

system	ca_ancora	cs_pcedt	cs_pdt	cu_proiel	de_parcorfull	de_potsdam	en_gum	en_litbank	en_parcorfull	es_ancora	fr_democrat	grc_proiel	hbo_ptnk	hu_korkor	hu_szeged	lt_lcc	no_bokmaalnarc	no_nynorsknarc	pl_pcc	ru_rucor	tr_itcc	average
CorPipe-2stage	82.22	**74.85**	**77.18**	**61.58**	69.53	71.79	**75.66**	**79.60**	68.89	**82.46**	68.16	**71.34**	**72.02**	63.17	**69.97**	**75.79**	**79.81**	**78.01**	**78.50**	**83.22**	**68.18**	73.90
CorPipe	81.02	73.71	75.84	60.72	**71.68**	71.45	74.61	79.10	**69.75**	80.98	**68.77**	68.53	70.86	60.32	68.12	75.78	79.55	77.52	77.03	83.09	59.37	72.75
CorPipe-single	80.42	72.82	74.82	57.11	61.62	67.02	74.39	78.08	58.61	79.75	67.89	66.01	67.18	60.09	67.32	75.19	78.92	76.60	75.20	81.21	53.43	70.18
Ours	**82.46**	70.82	75.80	54.97	71.40	**71.91**	70.53	74.15	55.58	81.94	62.69	61.64	61.56	**64.86**	69.26	71.97	74.51	72.07	76.34	80.47	64.49	69.97
baseline	68.32	64.06	63.83	24.51	47.21	55.65	63.19	63.54	33.08	69.58	53.62	28.76	24.60	35.14	54.51	62.00	64.96	63.70	66.24	65.83	44.05	53.16

Table 4: Results on test set.

$$J(D) = \log \prod_{i=1}^{N} \sum_{\hat{y} \in Y(i) \cap \texttt{GOLD}(i)} P(\hat{y}) + \underbrace{y_m^{(i)} \cdot \sigma(s_m(i)) + (1 - y_m^{(i)}) \cdot \sigma(-s_m(i))}_{\text{singletons binary cross-entropy}} \tag{2}$$

der if *CorPipe* uses any specific improvements to handle these languages better. Another possible explanation is that they were able to train large model models better, and large model handles these ancient languages with very little data available better.

5.1 Comparison To Other Systems

The comparison to other participating systems is shown in Table 4. Our system ended up in 4^{th} place (2^{nd} team). Surprisingly, although the winning system outperformed ours by a large margin on average, our system reached the best performance for three datasets (*german_potsdam*, *catalan*, and *hungarian-korkor*). It would be interesting to examine the differences between the two systems to find out why.

6 Conclusion

We further extended our system from CRAC 2022 and 2023 with the usage of mT5 through LORA training. We provide the analysis of different model configurations. We found out that for approximately half of the datasets, using a larger model does not help anymore. We also analyzed a drop caused by losing the gold annotation of empty nodes. Unfortunately, we did not have enough time to add zero nodes prediction into our model. Our results suggest that there is a lot of space for improvement. Our system ended up in 4^{th} place. Moreover, we reached the best performance on three datasets out of 21.

Acknowledgements

This work has been supported by by the Ministry of Education, Youth and Sports of the Czech Republic, Grant No. SGS-2022-016 Advanced methods of data processing and analysis. Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA CZ LM2018140) supported by the Ministry of Education, Youth and Sports of the Czech Republic.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard 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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

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 S 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.

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.

Anna Nedoluzhko, Michal Novák, Martin Popel, Zdeněk Žabokrtskỳ, Amir Zeldes, and Daniel Zeman. 2022. Corefud 1.0: Coreference meets universal dependencies. In *Proceedings of LREC*.

Anna Nedoluzhko, Michal Novák, Martin Popel, Zdeněk Žabokrtský, and Daniel Zeman. 2021. *Coreference meets Universal Dependencies – a pilot experiment on harmonizing coreference datasets for 11 languages*. ÚFAL MFF UK, Praha, Czechia.

Michal Novák, Barbora Dohnalová, Miloslav Konopík, Anna Nedoluzhko, Maciej Ogrodniczuk, 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*, Miami, Florida. 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.

Ondřej Pražák and Miloslav Konopík. 2024. Exploring multiple strategies to improve multilingual coreference resolution in corefud. *arXiv preprint arXiv:2408.16893*.

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.

Rhea Sukthanker, Soujanya Poria, Erik Cambria, and Ramkumar Thirunavukarasu. 2020. Anaphora and coreference resolution: A review. *Information Fusion*, 59:139–162.

Liyan Xu and Jinho D. Choi. 2020. Revealing the myth of higher-order inference in coreference resolution. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8527–8533, 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 Fifth Workshop on Computational Models of Reference, Anaphora and Coreference*, Gyeongju, Republic of Korea. Association for Computational Linguistics.