ÚFAL CorPipe at CRAC 2022: Effectivity of Multilingual Models for Coreference Resolution

Milan Straka and **Jana Straková**

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

Abstract

We describe the winning submission to the CRAC 2022 Shared Task on Multilingual Coreference Resolution. Our system first solves mention detection and then coreference linking on the retrieved spans with an antecedent-maximization approach, and both tasks are fine-tuned jointly with shared Transformer weights. We report results of fine-tuning a wide range of pretrained models. The center of this contribution are fine-tuned multilingual models. We found one large multilingual model with sufficiently large encoder to increase performance on all datasets across the board, with the benefit not limited only to the underrepresented languages or groups of typologically relative languages. The source code is available at `https://github.com/ufal/crac2022-corpipe`.

1 Introduction

Coreference resolution is a task of identifying and clustering multiple occurrences of entities across a textual document. The CRAC 2022 Shared Task on Multilingual Coreference Resolution (Žabokrtský et al., 2022) features coreference resolution on 13 datasets in 10 languages, originating from the CorefUD 1.0 multilingual dataset (Nedoluzhko et al., 2021, 2022).

Coreference resolution is often divided into two subtasks, *mention detection* and *coreference linking* (also *clustering*). Our contribution solves these tasks neither as a purely pipeline approach with two separate sequential models, nor as an end-to-end system (Lee et al., 2017, 2018), as is recently more common (e.g., the baseline; Pražák et al., 2021), but somewhere in between: We first solve mention detection and then coreference linking with an antecedent-maximization algorithm, but both tasks are jointly fine-tuned in one shared large language model. This circumvents the explosion of possible spans in an end-to-end approach, allows for a single retrieval of mentions only, while keeping the

benefit of sharing the weights and training only one model for two highly related tasks (contribution 1).

Our architecture is a fine-tuned large language model, with experimental results leaning toward large pretrained models with better multilingual representation. We experimented with a wide range of pretrained language models (Devlin et al., 2019; Conneau et al., 2020; Chung et al., 2021; Armengol-Estapé et al., 2021; Straka et al., 2021; Chan et al., 2020; Devlin et al., 2019; Joshi et al., 2020; Cañete et al., 2020; Martin et al., 2020; Nemeskey, 2020; Ulčar and Robnik-Šikonja, 2021; Mroczkowski et al., 2021; Kuratov and Arkhipov, 2019), of which RemBERT (Chung et al., 2021) proved the most effective (contribution 2).

We found *multilingual models* at the center of our research attention in the CRAC 2022 Shared Task. The shared task featured datasets with sizes ranging from tiny (457 training sentences in `de` and `en parcorfull`) to relatively large (nearly 40K training sentences in `cs pcedt` and `cs pdt`), all of them evaluated with equal weight (macro average). This implied that special care must be devoted to leveling the performance on all datasets. We experimented with various combinations of fine-tuned multilingual models and various sampling strategies. Although our motivation was to mitigate the poor performance on smaller specimens, we surprisingly found that one large multilingual model with sufficiently large encoder improves results on all datasets, not only the small or linguistically related ones (contribution 3).

To sum up, our contributions are the following:

1. We present a jointly trained pipeline approach for coreference resolution.
2. Although many monolingual base models surpass their multilingual base counterparts, in the end, one large multilingual pretrained model gives the best performance over base, albeit specifically pretrained monolingual encoders.

3. One fine-tuned all-data multilingual model with sufficiently large encoder outperforms individual models across all datasets, not only the smaller or typologically related ones.

The source code of our system is available at https://github.com/ufal/crac2022-corpipe.

2 Related Work

Coreference resolution is often divided into two subtasks: *mention detection* and *coreference linking* (or *clustering*). These can be solved either separately (pipeline approach) or, more recently, in an end-to-end fashion (Lee et al., 2017, 2018). Such was also the approach of the baseline (Pražák et al., 2021). Our proposal takes what we hope is advantageous from both approaches: We solve both tasks sequentially, but the weights are trained jointly in a shared network.

As in all other NLP areas, deep learning with representations from large language models represents the current state-of-the-art (Kantor and Globerson, 2019; Joshi et al., 2019, 2020). We build on these works which use BERT (Devlin et al., 2019) by comparing BERT with its successors, the language-specific mutations of BERT (Armengol-Estapé et al., 2021; Straka et al., 2021; Chan et al., 2020; Devlin et al., 2019; Joshi et al., 2020; Cañete et al., 2020; Martin et al., 2020; Nemeskey, 2020; Ulčar and Robnik-Šikonja, 2021; Mroczkowski et al., 2021; Kuratov and Arkhipov, 2019), and multilingual variants: mBERT (Devlin et al., 2019), XLM-R base and XLM-R large (Conneau et al., 2020), and RemBERT (Chung et al., 2021).

There are mixed accounts in the literature on globally decoding the entities (clusters) via higher-order methods. Kantor and Globerson (2019) improved state-of-the-art on the CoNLL-2012 shared task with differentiable end-to-end manner enabling higher-order inference: mentions are represented as the sum of all mentions of the entity (*entity equalization*). Other higher-order coreference linking methods include *attended antecedent* (Lee et al., 2018; Fei et al., 2019; Joshi et al., 2019, 2020). On the other hand, Xu and Choi (2020) thoroughly investigated the contribution of higher-order methods to the models performance and conclude that with modern encoders, higher-order methods contribute only marginally or negatively. As we model *all* antecedent links during training in a dot-product attention matrix, we inherently "equalize" entities, although not with an explicit algorithm.

3 Methods

An overview of the model architecture is shown in Figure 1. In the following sections, we describe the components of the model in detail, with reference to the corresponding parts of Figure 1.

3.1 Architecture

We consider that the enumeration of all possible spans as mention candidates in an end-to-end approach, despite aggressive pruning, may lead to explosion of options and possibly harm the coreference linking because the candidate set is heavily biased toward negative outcome: only a fraction of the spans is an actual mention and of these, only a fraction is a mention of the same entity. Furthermore, this approach does not allow the retrieval of mentions only. Hence, we propose a jointly trained, pipeline approach: we first solve *mention detection* and then *coreference linking* only on the retrieved mentions. However, to share the information between these highly related tasks and to keep a single model, we fine-tune one shared large language model, only with separately stacked hidden layers on top of the shared large language model for each task. In Figure 1, the orange box corresponds to the shared fine-tuned large language model (encoder), the green box corresponds to the *mention detection* task and the purple box corresponds to the *coreference linking* task.

In all fairness it should be said that we did not experimentally compare our architecture with the purely pipeline models with separate encoders nor the end-to-end approach, as pursuing three architectures to final submission was beyond our means in the given time frame. We venture to suggest that separately trained pipeline models might have the advantage of greater capacity (separate large language model for each task) but might become expensive as both models must be separately fine-tuned and hyperparameter-searched.

3.2 Token Representations (Encoder)

Each token receives a contextualized representation from the encoder, a vector of dimension D ($D = 768$ for base encoders, $D = 1024$ for XLM-R large, $D = 1152$ for RemBERT). The retrieval of contextualized token representation corresponds to the orange box in the bottom of Figure 1. The representation is shared between the *mention detection* and *coreference linking* tasks.
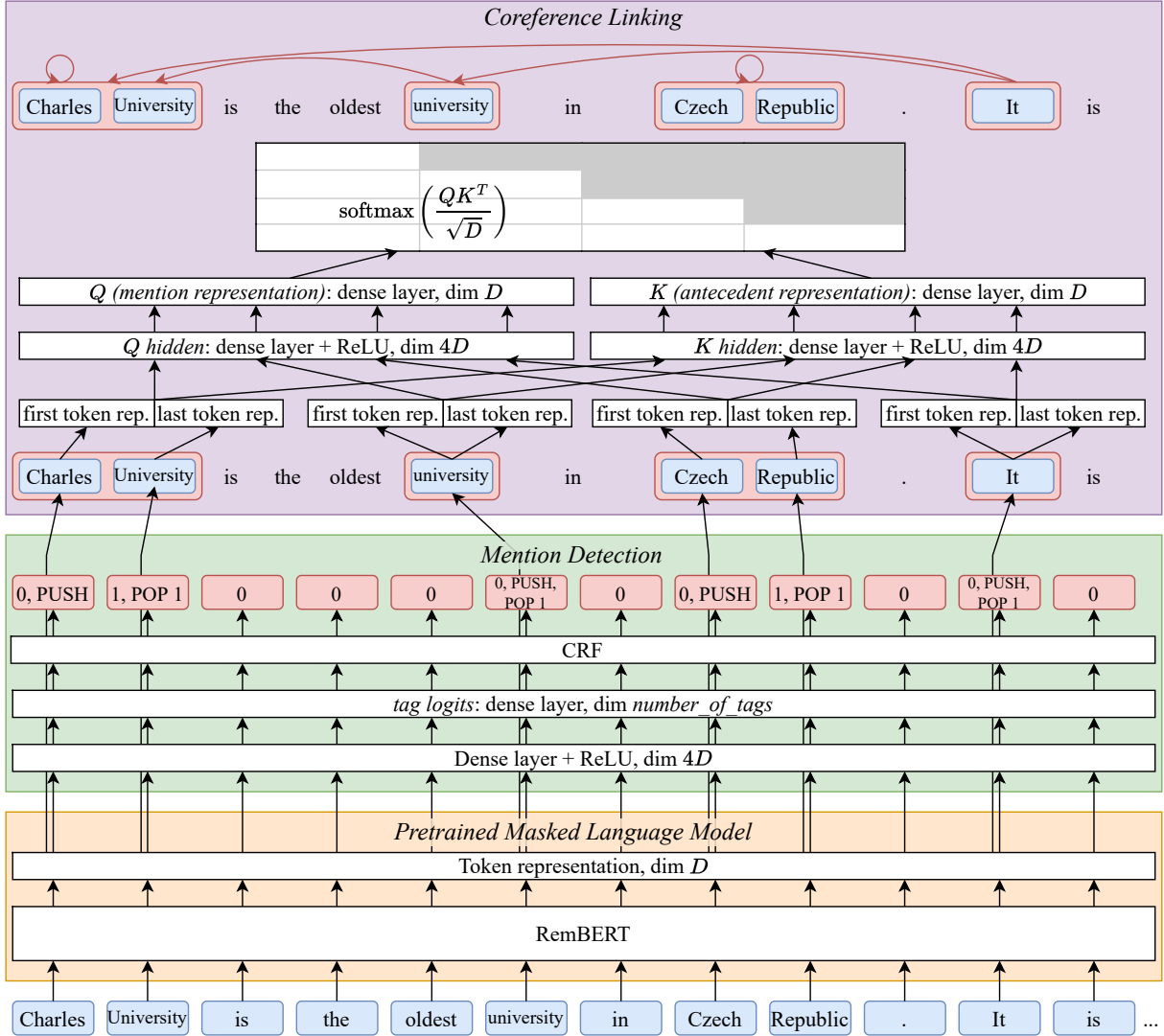
Figure 1: CorPipe model architecture. Best viewed in color.

We experimented with the following pretrained multilingual language models:

- RemBERT (Chung et al., 2021),
- XLM-R base and large (Conneau et al., 2020),
- mBERT (Devlin et al., 2019),

and the following published language-specific models:[1]

- Catalan BERTa (Armengol-Estapé et al., 2021),
- Czech RoBERTa RobeCzech (Straka et al., 2021),
- German gBERT (Chan et al., 2020),
- English SpanBERT (Joshi et al., 2020),
- Spanish BETO (Cañete et al., 2020),
- French CamemBERT (Martin et al., 2020),

- Hungarian HuBERT (Nemeskey, 2020),
- Lithuanian LitLatBERT (Ulčar and Robnik-Šikonja, 2021),
- Polish HerBERT (Mroczkowski et al., 2021),
- Russian RuBERT (Kuratov and Arkhipov, 2019).

3.3 Empty Nodes

Some dependency grammar annotation schools allow, or even require, the so-called *empty nodes*, which are superficial nodes of a dependency graph unseen on the surface level, i.e., not directly corresponding to any surface token of the sentence. The empty nodes usually account for ellipsis, such as in a sentence *"Mary likes roses and John (likes) violets."*, in which the verb "likes" is omitted but depending on the annotation guidelines may be reconstructed in the dependency tree. These nodes

---

[1]For each language, we present the monolingual model that worked best in our settings, if more exist.

may carry coreference annotation, and they very often do in pro-drop languages (like Slavic or Romance languages). In Czech example: *"Řekl, že nepřijde."*, translated as *"(He) said that (he) won't come."*, both pronouns are dropped but implied by the morphology of the verb.

To allow the empty nodes, if occurring, to be naturally represented on the input and the output of the fine-tuned model and be part of the fine-tuning, we simply draw them to the surface, that is, we create a new token occupying the implied position of the artificial empty node and assign whatever text that was annotated with it (or none).[2] To recognize such artificial tokens from regular tokens, we prepend an artificial special character to any such token originating from an empty-node.

3.4 Mention Detection

We model mention detection as a sequence token-level classification problem, which considers a sequence of tokens on the input and a corresponding sequence of tags on the output. The proposed tags are an extension of BIO encoding, which in addition can handle embedded and also overlapping mention spans. Each tag is a sequence of the following stack manipulation instructions:

- $0..N$ POP instructions, each closing a mention from the stack. To handle crossing mention spans, the instruction has a parameter specifying which mention to close using its index from the top of the stack. The most frequently used value is 1 (the top of the stack), because closing the mention on the top of the stack is sufficient to encode arbitrarily embedded non-crossing mention spans.
- $0..N$ PUSH instructions, each starting a new mention on the top of the stack.
- $0..N$ POP instructions again, each closing a single-token mention started by a previous PUSH in the same step. We could represent such single-word mentions using specialized UNIT instructions instead of a PUSH-POP pair, but we opted for less instructions for the simplicity of the decoder.

The above mentioned stack instructions are concatenated into a single tag, predicted by a classifier as one label per token.

Because not all sequences of tags are valid (i.e., we are performing structured prediction), we pro-

cess the tags by a linear-chain CRF. Finally, in order to allow the CRF to check whether there is a mention to be closed by a POP instruction, we include the size of the stack in the tag.[3]

The mention detection classifier corresponds to the green box in Figure 1. Token representation of dimension D is processed by a hidden ReLU layer of dimension $4D$, then by a linear layer producing tag logits, and finally by a CRF layer.

3.5 Coreference Linking

We approach coreference linking by considering, for each mention, a probability distribution of the preceding mentions in the previous context (more on context window in Section 3.6) being antecedents of the current mention. We also include the mention itself in the distribution, and consider it a technical antecedent if the mention has no antecedents.

During training, our goal is to predict *all* mention antecedents using a categorical cross-entropy loss. During prediction, however, we predict only the most probable antecedent for every mention, noting that any correct antecedent results in the same coreference cluster.[4]

The computation of the antecedent distribution, corresponding to the purple box in Figure 1, starts by constructing an initial representation of every mention by concatenating the token representations of its first and last tokens.[5] Using this representation, we compute Q (the representation of a reference candidate) and K (the representation of an antecedent candidate), both using a hidden ReLU layer with dimensionality $4D$ followed by a bias-free linear layer of dimensionality D. Finally, we compute the antecedent distribution using masked dot-product self-attention (Vaswani et al., 2017).

The inclusion of "self" in the pool of antecedents naturally allows for the so-called *singletons*, which are mentions without antecedent (entities mentioned only once, for example "Czech Republic" in Figure 1). Singletons were excluded from the

---

[2]We use the form associated with a given empty node; if empty, we fall back to the (possibly empty) lemma.

[3]Our approach does not handle discontinuous mentions. While we could support them by introducing an instruction continuing an already closed span, handling discontinuous mentions would also require support in the mention encoder.

[4]This is true only when considering previous mentions as antecedents; if we considered both previous and following mentions as antecedents, disconnected components of a single coreference cluster could be formed.

[5]Such an approach assumes the mentions are continuous. We handle discontinuous mentions by limiting them to their largest continuous sub-span containing the syntactic head of the mention (see Section 3.8).

official evaluation primary metric, but the official evaluation with singletons on the test set and the ablation experiments with singletons on the dev set can be found in Section 4.6.

The fact that during training a reference should recognize all its antecedents might seem inconsistent with the inference regime, where only a single most probable antecedent is retrieved. We therefore demonstrate the effectiveness of considering all antecedents by also evaluating a strategy of limiting the number of gold antecedents to at most 1, 2, or 3 previous ones (*At most 1 link*, *At most 2 links* or *At most 3 links*) in Section 4.5.

3.6 Context Window

For each sentence, we consider a sliding context of 512 tokens, aligning the end of the current sentence towards the end of the window to allow for a larger left (past) context than the right (future) context. We experiment with several settings of the size of the right context in Section 4.4:

- **Right context 0**: The end of the current sentence is perfectly aligned with the context of 512 tokens (no right context).
- **Right context 50**: We leave 50 tokens for the right (future) context after the sentence end and whatever remains is the left (past) context; if there is not enough left context to fill the whole window of 512 tokens (e.g., the first sentence of the document), we increase the size of the right context to fill all the 512 tokens.
- **Right context 100**: Same as before, but 100 tokens for the right context.

Unless stated otherwise, we use right context of 50.

3.7 Multilingual Models

We introduced *multilinguality* as our natural research interest of CRAC 2022 Shared Task. We experimented with various combinations of models with respect to size and/or language, and in the end, we submitted three contributions to the final evaluation:

- **individual**: The models were fine-tuned using solely the training data of the corresponding dataset.[6]
- **multilingual**: All training data were used for fine-tuning a single multilingual model, with

---

[6]With the exception of de and en parcorfull – these corpora are extremely small (457 sentences each) and translations of each other, so we always train on a concatenation of them when finetuning an individual language model.
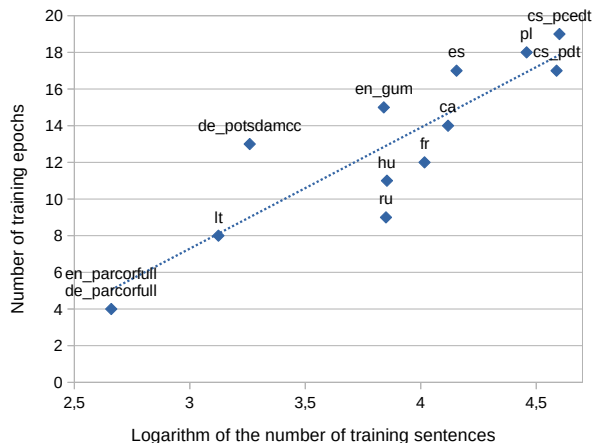

Figure 2: Dependency of the number of the optimum training epochs on the logarithm of the corpus size.

examples sampled according to the logarithm of the individual dataset sizes. The final checkpoint of the last training epoch was used for prediction, so there is only one single large model for all datasets, an option which could most easily qualify as a deliverable software product for multilingual coreference resolution.

- **best dev**: In this setting we considered, for every dataset, the test set prediction corresponding to a model and optimum epoch achieving the best development set performance. An intuition behind this decision is that in the multilingual settings, the smaller datasets converge sooner, while the large ones need more iterations. This is supported by the seemingly linear relationship between the logarithm of the number of training sentences and the optimum number of training epochs in Figure 2.

When mixing multilingual data, sampling ratios of the datasets must be determined. We experimented with three strategies for sampling the datasets; the examples are then always sampled uniformly from the chosen dataset:

- **logarithmic**: Datasets are sampled with the probability reflecting the logarithm of their size.[7]
- **uniform**: Datasets are sampled with uniform probability.
- **linear**: Datasets are sampled in proportion linear to their size, which effectively equals to sampling the examples uniformly from the concatenation of the datasets.

---

[7]We scaled the logarithmic sampling rations to the range 1 to 5, and rounded them for convenience.

Finally, dataset labels (corpus ids) may or may not be be added to the input to discriminate the origins. We call these settings *w/ corpus id* and *w/o corpus id*.

We compare all the above mentioned strategies for creating multilingual models in Section 4.1.

3.8 Limiting Mention Spans to Their Heads

The official CRAC 2022 Shared Task evaluation relied on the lenient *partial matching*, which considers mention span correctly detected if it contains the syntactic head of the gold mention and at the same time, the predicted mention span does not include any tokens outside the gold mention span. Hence it seems prudent to not "overpredict" too long mention spans and prune the predicted mention spans to their syntactic head, given that syntactic analysis is available in the data. We show ablation results including the full mention spans in Section 4.7.

3.9 Training

We trained our models using a lazy variant of the Adam optimizer (Kingma and Ba, 2015), with a batch size of 8. The *base* variants were fine-tuned on a single 16GB GeForce/Quadro GPU, using a slanted triangular learning rate schedule – first linearly increasing from 0 to $2 \cdot 10^{-5}$ in the first 10% of the training, and then linearly decaying to 0 at the end of the training. The multilingual models were trained for 30 epochs, each consisting of 6000 batches; the individual models were trained for up to 100 epochs depending on dataset size.

The *large* models required fine-tuning on two 25GB GeForce GPUs, the peak learning rate was 10^{-5}, the multilingual models were trained for 20 epochs and the individual models up to 50 epochs. We trained 8 large multilingual models (each taking 42 hours), considering both XLM-R large and RemBERT, uniform and logarithmic mixing, presence of corpus id, and $\beta_2 = 0.99$ in addition to the default one. The best-performing model uses RemBERT, logarithmic mixing without corpus id, and default β_2.

4 Results

Official results of the CRAC 2022 Shared Task on the test set can be found in Table 1. Our multilingual models, *best dev* and *multilingual*, scored 1st and 2nd, respectively, while our *individual* models trained on each dataset placed 4th.

4.1 Multilingual Models

A view on the effectiveness of multilingual models is shown in official ablation results on test data in Table 2, which compares all our three individual/multilingual settings: *multilingual* as a baseline, *individual* and *best dev*, using a base encoder (XLM-R base for the multilingual baseline, best-performing base encoder for the remaining cases) and a large encoder (RemBERT). The *multilingual* is superior to *individual* for all datasets, with the exception of the three largest datasets using a base encoder – we hypothesize that the base encoder does not have sufficient capacity to capture the largest datasets in the *multilingual* setting, because with a large encoder, also the three largest datasets benefit from the *multilingual* model. Furthermore, Table 4.C demonstrates that while XLM-R large is the best in the *individual* settings, RemBERT delivers superior *multilingual* performance.

Motivated by the improvements of the multilingual models, we considered a setting where 50% of the training data comes from a single dataset and the rest from all other datasets (with logarithmic mixing). Surprisingly, such setting delivers consistently worse performance than the multilingual models (last line of Table 4.C).

The comparison of *logarithmic*, *uniform*, and *linear* mixing, together with the presence or absence of *corpus id*, is evaluated in Table 4.D and Table 4.E. Unexpectedly, neither the mixing rations nor the *corpus id* have a large effect on the results, which is surprising especially for the *linear* mixing, where the smallest treebanks are nearly 100 times less frequent than the largest one.

4.2 Zero-shot Evaluation

The prospect of not including the corpus id opens an interesting possibility of using the model in zero-shot setting, i.e., on a different language than it was trained on. To perform such zero-shot evaluation, we trained for every language a multilingual model *without* datasets in this language, and then evaluated the model on them. The results, presented in Table 4.F, were below our expectations, slightly surpassing 60% macro average on the development set with the RemBERT model.

4.3 Monolingual Pretrained Language Models

Table 4.G presents the evaluation of the best-performing monolingual base-sized pretrained models we found. While the specialized models

Team/Submission	Avg	ca	cs pcedt	cs pdt	de parc	de pots	en gum	en parc	es	fr	hu	lt	pl	ru
ÚFAL CorPipe *best dev*	**70.72** (1)	78.18 (2)	**78.59** (1)	**77.69** (1)	65.52 (2)	70.69 (2)	72.50 (2)	**39.00** (1)	**81.39** (1)	**65.27** (1)	63.15 (3)	**69.92** (1)	78.12 (2)	**79.34** (1)
ÚFAL CorPipe *multilingual*	69.56 (2)	**78.49** (1)	78.49 (2)	77.57 (2)	59.94 (3)	**71.11** (1)	**73.20** (1)	33.55 (3)	80.80 (2)	64.35 (3)	63.38 (2)	67.38 (3)	**78.32** (1)	77.74 (2)
UWB *ondfa†*	67.64 (3)	70.55 (4)	74.07 (4)	72.42 (4)	**73.90** (1)	68.68 (3)	68.31 (4)	31.90 (4)	72.32 (4)	61.39 (4)	**65.01** (1)	68.05 (2)	75.20 (4)	77.50 (3)
ÚFAL CorPipe *individual*	64.30 (4)	76.34 (3)	77.87 (3)	76.76 (3)	36.50 (5)	56.65 (5)	70.66 (3)	23.48 (5)	78.78 (3)	64.94 (2)	62.94 (4)	61.32 (6)	73.36 (5)	76.26 (4)
Barbora Dohnalová *berulasek*	59.72 (5)	64.67 (5)	70.56 (5)	67.95 (5)	38.50 (4)	57.70 (4)	63.07 (5)	36.44 (2)	66.61 (5)	56.04 (5)	55.02 (5)	65.67 (4)	65.99 (6)	68.17 (5)
UWB BASELINE‡	58.53 (6)	63.74 (6)	70.00 (6)	67.27 (6)	33.75 (6)	55.44 (6)	62.59 (6)	36.44 (2)	65.98 (6)	55.55 (6)	52.35 (6)	64.81 (5)	65.34 (7)	67.66 (6)
Matouš Moravec *moravec*	55.05 (7)	58.25 (7)	68.19 (7)	64.71 (7)	31.86 (7)	52.84 (7)	59.15 (7)	36.44 (2)	62.01 (7)	54.87 (7)	52.00 (7)	59.49 (7)	63.40 (8)	52.49 (7)

Table 1: Official results of CRAC 2022 Shared Task on the test set (CoNLL score in %). The systems † and ‡ are described in Pražák and Konopik (2022) and Pražák et al. (2021), respectively; the rest in Žabokrtský et al. (2022).

Experiment	Avg	ca	cs pcedt	cs pdt	de parc	de pots	en gum	en parc	es	fr	hu	lt	pl	ru
XLM-R base, multilingual	67.8	77.1	75.8	74.3	54.7	66.9	70.1	38.5	77.6	64.2	62.3	69.4	73.3	76.6
Best base model, individual	-5.2	-4.0	+1.5	+2.2	-18.2	-9.8	-3.4	-15.0	-2.4	-2.4	-2.0	-8.1	+0.0	-5.6
Best base model, best dev	+0.4	-0.6	+1.5	+2.2	+2.0	+0.6	-1.0	-0.9	+1.2	-1.1	+0.6	+0.4	+0.0	+0.2
RemBERT, multilingual	+1.8	**+1.4**	+2.6	+3.3	+5.2	**+4.2**	**+3.1**	-4.9	+3.2	+0.1	**+1.1**	-2.0	**+5.0**	+1.1
RemBERT, individual	-3.5	-0.7	+2.0	+2.5	-18.2	-10.3	+0.6	-15.0	+1.2	+0.7	+0.7	-8.1	+0.0	-0.4
RemBERT, best dev	**+3.0**	+1.1	**+2.7**	**+3.4**	**+10.8**	+3.8	+2.4	**+0.5**	**+3.8**	**+1.0**	+0.9	**+0.5**	+4.8	**+2.7**

Table 2: Official results of ablation experiments on the test set (CoNLL score in %).

Team/Submission	Avg. with singletons
ÚFAL CorPipe, *best dev*	**72.98**
ÚFAL CorPipe, *multilingual*	71.81
ÚFAL CorPipe, *individual*	67.93
UWB, *ondfa*	58.06
Barbora Dohnalová, *berulasek*	50.84
UWB, BASELINE	49.69
Matouš Moravec, *moravec*	46.79

Table 3: Official results of evaluation with singletons on the test set.

consistently surpass mBERT and are mostly better than XLM-R base, they are all worse than the individual XLM-R large models (with the exception of Lithuanian) and even more dominated by the RemBERT multilingual model. This indicates that, nowadays, pretraining a base-sized monolingual BERT model has merit only in improving the running time, not model performance, when large pretrained multilingual models are now available.

4.4 Context Window

Table 4.H shows the effect of using a right context of size 0, 50, and 100. The evaluation, performed on a base-sized model with a preliminary, develop-

ment version of CorPipe, shows that the presence of the right context is beneficial, but does not clearly indicate whether context of size 100 is better than 50.

4.5 Number of Links

The effect of limiting the number of predicted antecedents during training is presented by Table 4.I. The evaluation (performed again on a base-sized model with a preliminary, development version of CorPipe) shows that performance increases with the number of antecedents considered during training.

4.6 Singletons

Singletons (entities with only one mention in the document) were excluded from the official evaluation primary metric. Our antecedent-maximization strategy however accounts for them by adding "self" to antecedent candidates pool. We publish the official evaluation with singletons on the test set in Table 3 and the ablation evaluation with singletons on the dev set in Table 4.B.

Experiment	Avg	ca	cs pcedt	cs pdt	de parc	de pots	en gum	en parc	es	fr	hu	lt	pl	ru
A) THE EFFECT OF USING FULL MENTIONS INSTEAD OF ONLY THEIR HEADS														
CorPipe multilingual	73.2	76.9	79.3	78.1	70.6	74.7	74.8	61.2	80.9	67.4	64.6	76.0	75.2	71.4
+ full mentions	-1.8	-2.4	-1.2	-0.9	-3.0	-2.1	-1.7	-3.0	-2.6	-1.1	-1.8	-1.3	-1.2	-1.2
CorPipe individual	71.1	76.3	78.7	76.9	65.7	62.0	73.8	63.2	79.5	66.8	64.8	73.4	71.7	69.9
+ full mentions	+0.3	-1.9	-0.7	+0.2	+1.9	+10.6	-0.7	-5.0	-1.3	-0.5	-1.9	+1.3	+2.3	+0.3
CorPipe best dev	**76.0**	**78.1**	**79.5**	**78.5**	**73.9**	**78.3**	**76.0**	**75.1**	**81.8**	**69.0**	**69.2**	**78.0**	**76.3**	**74.6**
+ full mentions	-4.6	-3.6	-1.4	-1.3	-6.3	-5.6	-3.0	-16.9	-3.5	-2.6	-6.3	-3.3	-2.4	-4.4
B) EVALUATION INCLUDING SINGLETONS														
CorPipe multilingual	74.8	82.3	78.0	74.5	68.7	79.8	82.2	52.1	85.1	76.6	63.3	73.7	84.2	69.5
CorPipe individual	73.7	82.4	77.2	73.4	63.4	71.7	82.2	60.6	84.1	76.4	62.5	71.3	82.2	67.8
CorPipe best dev	**77.5**	**83.2**	**78.0**	**74.7**	**70.4**	**82.6**	**83.2**	**71.6**	**85.8**	**77.5**	**66.8**	**74.8**	**84.7**	**73.0**
C) EFFECT OF MULTILINGUAL DATA AND THE PRETRAINED MODEL														
XLM-R base multilingual	73.3	75.8	76.0	75.0	73.4	74.1	73.1	**75.4**	78.4	66.1	65.2	**78.0**	72.1	71.7
XLM-R large multilingual	+1.5	**+1.7**	+1.8	+2.0	+0.3	**+4.1**	+2.1	-4.5	+2.2	+1.7	+3.1	-0.0	+2.9	+0.9
RemBERT multilingual	**+1.9**	+1.6	**+3.3**	**+3.3**	**+2.9**	+2.4	**+2.4**	-6.1	**+2.7**	**+2.0**	**+4.0**	-1.2	**+3.7**	**+2.9**
XLM-R base individual	-4.6	-4.4	-0.3	-1.1	-7.8	-12.1	-1.9	-12.2	-2.8	-3.0	-3.8	-4.6	-2.3	-6.1
XLM-R large individual	-0.6	+0.2	+2.8	+3.0	-7.7	-5.2	-0.9	-4.4	+1.0	+0.3	+3.7	-5.4	+3.5	-1.2
RemBERT individual	-4.7	+0.6	+2.8	+1.9	-23.0	-12.1	+0.7	-30.5	+1.1	+0.7	-0.4	-8.9	+2.7	-1.8
RemBERT 50% additional	+0.3	+1.0	+2.5	+2.4	-1.4	-0.5	+1.7	-8.3	+0.9	+1.3	+1.6	-3.5	+3.6	+1.7
D) EFFECT OF MIXING RATIOS using XLM-R base Pretrained Model														
Logarithmic, w/o corpus id	**73.3**	75.8	76.0	75.0	**73.4**	**74.1**	**73.1**	**75.4**	78.4	66.1	65.2	78.0	72.1	**71.7**
Logarithmic, w/ corpus id	-0.4	-0.5	+0.1	+0.3	-0.8	-0.3	-0.6	-4.6	**+0.1**	**+0.6**	**+1.3**	-0.9	**+0.3**	-0.7
Uniform, w/o corpus id	-0.8	-0.5	-0.2	-0.9	-1.8	-3.5	-0.2	-1.9	+0.0	-0.0	+0.4	-1.1	+0.0	-1.5
Uniform, w/ corpus id	-1.6	-1.1	-0.5	-0.6	-6.4	-2.1	-0.4	-7.0	+0.1	+0.1	-0.5	-1.1	-0.6	-1.2
Linear, w/o corpus id	-0.3	**+0.1**	**+0.8**	**+1.1**	-1.1	-0.5	-0.5	-3.5	-0.1	+0.3	+1.0	**+0.3**	-0.1	-1.6
E) EFFECT OF MIXING RATIOS using RemBERT Pretrained Model														
Logarithmic, w/o corpus id	75.3	77.4	79.3	78.3	76.3	76.5	75.5	69.3	81.1	68.1	**69.2**	76.8	75.8	**74.6**
Logarithmic, w/ corpus id	**+0.6**	+0.4	+0.1	+0.1	**+3.0**	**+1.2**	-0.1	**+5.8**	+0.3	**+0.9**	-2.4	-1.3	+0.1	-0.2
Uniform, w/o corpus id	+0.1	+1.2	-0.3	-0.1	+2.4	+0.5	**+0.0**	-0.9	-0.1	+0.7	-0.6	-0.2	**+0.1**	-1.2
Uniform, w/ corpus id	-0.1	-0.0	-0.2	-0.3	-4.2	+0.3	-0.1	+4.5	+0.4	+0.6	-1.0	-0.1	+0.1	-1.2
Linear, w/o corpus id	-0.1	**+1.3**	**+0.1**	**+0.2**	-2.3	-0.5	-1.5	+1.9	**+0.5**	+0.7	-1.0	**+0.4**	+0.0	-1.3
F) ZERO-SHOT EVALUATION OF A MULTILINGUAL MODEL														
Multilingual XLM-R base	73.3	75.8	76.0	75.0	73.4	74.1	73.1	**75.4**	78.4	66.1	65.2	**78.0**	72.1	71.7
Zero-shot XLM-R base	-17.1	-11.1	-28.6	-23.8	-13.3	-13.8	-19.8	-18.5	-6.8	-7.6	-16.1	-23.8	-24.6	-15.1
Multilingual RemBERT	**+1.9**	**+1.6**	**+3.3**	**+3.3**	**+2.9**	**+2.4**	**+2.4**	-6.1	**+2.7**	**+2.0**	**+4.0**	-1.2	**+3.7**	**+2.9**
Zero-shot RemBERT	-12.5	-6.7	-23.7	-20.6	-11.1	-7.5	-15.6	-9.8	-2.8	-8.3	-10.5	-20.0	-18.3	-7.2
G) EFFECT OF SEVERAL LANGUAGE-SPECIFIC base Pretrained Models														
XLM-R base individual	68.7	71.4	75.7	73.9	65.7	62.0	71.2	63.2	75.6	63.1	61.5	73.4	69.8	65.6
mBERT (Devlin et al., 2019)	-2.8	-1.5	-3.0	-3.4	-3.3	+0.4	-2.8	-1.1	-1.8	-1.1	-2.7	-7.5	-4.4	-3.6
BERTa (Armengol-Estapé et al., 2021)		+1.3												
RobeCzech (Straka et al., 2021)			+2.0	+2.8										
gBERT (Chan et al., 2020)					-9.9	+5.3								
SpanBERT (Joshi et al., 2020)							-0.4	-2.4						
BETO (Cañete et al., 2020)									+0.4					
CamemBERT (Martin et al., 2020)										-0.2				
HuBERT (Nemeskey, 2020)											+3.6			
LitLatBERT (Ulčar and Robnik-Šikonja, 2021)												+2.7		
HerBERT (Mroczkowski et al., 2021)													+1.6	
RuBERT (Kuratov and Arkhipov, 2019)														+0.2
XLM-R large individual	+4.0	+4.6	+3.1	+4.1	+0.0	+6.9	+1.0	**+7.8**	+3.8	+3.3	+7.4	-0.8	+5.8	+4.8
RemBERT individual	-0.0	+4.9	+3.1	+3.1	-15.2	+0.0	+2.6	-18.3	+3.9	+3.8	+3.3	-4.3	+5.0	+4.3
XLM-R large multilingual	+6.1	**+6.1**	+2.1	+3.2	+8.0	**+16.2**	+4.1	+7.7	+5.0	+4.8	+6.9	**+4.6**	+5.1	+6.9
RemBERT multilingual	**+6.6**	+6.0	**+3.6**	**+4.4**	**+10.6**	+14.5	**+4.3**	+6.1	**+5.5**	**+5.1**	**+7.7**	+3.5	**+6.0**	**+9.0**
H) EFFECT OF THE RIGHT CONTEXT SIZE; DEVELOPMENT VERSION														
Right context 0	67.4	**70.7**	75.0	73.6	59.2	62.4	**68.3**	68.6	**74.4**	61.1	59.2	71.5	69.2	62.2
Right context 50	**+0.8**	-0.4	+1.3	+0.6	+3.9	+1.8	-0.7	**+1.2**	-0.5	+0.0	+1.7	**+0.5**	+0.3	**+1.7**
Right context 100	+0.6	-1.2	**+1.5**	**+0.7**	**+5.1**	**+2.7**	-0.1	-3.8	-0.5	**+0.6**	**+2.1**	-0.2	**+0.6**	+0.7
I) EFFECT OF THE MAXIMUM NUMBER OF LINKS DURING TRAINING; DEVELOPMENT VERSION														
Unlimited	**67.4**	**70.7**	75.0	**73.6**	59.2	**62.4**	68.3	**68.6**	74.4	**61.1**	**59.2**	71.5	**69.2**	62.2
At most 1 link	-3.8	-3.3	-0.9	-3.6	-3.1	-4.3	-4.8	-8.7	-4.0	-3.1	-2.6	-5.0	-3.0	-3.8
At most 2 links	-1.4	-1.4	+0.2	-2.0	+1.5	-2.4	-1.9	-5.6	-0.8	-0.1	-1.0	-0.5	-2.6	-1.3
At most 3 links	-0.6	-0.9	**+0.5**	-0.2	**+3.5**	-0.4	-0.4	-6.5	-1.2	-0.4	-0.2	**+1.1**	-1.9	-0.6

Table 4: Ablation experiments evaluated on the development sets (CoNLL score in %). In A) and B), the scores of the official submissions are used; in C) to I), we report the highest development set score from any epoch.

4.7 Limiting Mention Spans to Their Heads

Comparison between full predicted mention spans and the predicted spans reduced to their syntactic heads in Table 4.A shows that *partial matching* favors post-processing which keeps syntactic heads and avoids "overprediction" beyond the gold mention span.

5 Conclusions

We presented a jointly trained pipeline approach as a winning contribution to the CRAC 2022 Shared Task on Multilingual Coreference Resolution (Žabokrtský et al., 2022). We published a thorough comparison of pretrained large language models for the task. Finally, we focused on multilingual models and we conclude that one multilingual, all-data model with large encoder outperformed individual monolingual fine-tuned models on all datasets. The source code is available at `https://github.com/ufal/crac2022-corpipe`.

References

Jordi Armengol-Estapé, Casimiro Pio Carrino, Carlos Rodriguez-Penagos, Ona de Gibert Bonet, Carme Armentano-Oller, Aitor Gonzalez-Agirre, Maite Melero, and Marta Villegas. 2021. Are multilingual models the best choice for moderately under-resourced languages? A comprehensive assessment for Catalan. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4933–4946, Online. Association for Computational Linguistics.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking embedding coupling in pre-trained language models. In *International Conference on Learning Representations*.

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.

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

Hongliang Fei, Xu Li, Dingcheng Li, and Ping Li. 2019. End-to-end deep reinforcement learning based coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 660–665, Florence, Italy. 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.

Ben Kantor and Amir Globerson. 2019. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677, Florence, Italy. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster), CoRR abs/1412.6980*.

Yuri Kuratov and Mikhail Y. Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. *CoRR*, abs/1905.07213.

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.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Robert Mroczkowski, Piotr Rybak, Alina Wróblewska, and Ireneusz Gawlik. 2021. HerBERT: Efficiently pretrained transformer-based language model for Polish. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 1–10, Kiyv, Ukraine. Association for Computational Linguistics.

Anna Nedoluzhko, Michal Novák, Martin Popel, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Peter Bourgonje, Silvie Cinková, Jan Hajič, Christian Hardmeier, Pauline Krielke, Frédéric Landragin, Ekaterina Lapshinova-Koltunski, M. Antònia Martí, Marie Mikulová, Maciej Ogrodniczuk, Marta Recasens, Manfred Stede, Milan Straka, Svetlana Toldova, Veronika Vincze, and Voldemaras Žitkus. 2022. Coreference in universal dependencies 1.0 (CorefUD 1.0). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Anna Nedoluzhko, Michal Novák, Martin Popel, Zdeněk Žabokrtský, and Daniel Zeman. 2021. Coreference in universal dependencies 0.1 (CorefUD 0.1). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Dávid Márk Nemeskey. 2020. *Natural Language Processing Methods for Language Modeling*. Ph.D. thesis, Eötvös Loránd University.

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 Konopik. 2022. End-to-end Multilingual Coreference Resolution with Mention Head Prediction. In *Proceedings of the CRAC 2022 Shared Task on Multilingual Coreference Resolution*. Association for Computational Linguistics.

Milan Straka, Jakub Náplava, Jana Straková, and David Samuel. 2021. Robeczech: Czech roberta, a monolingual contextualized language representation model. In *24th International Conference on Text, Speech and Dialogue*, pages 197–209, Cham, Switzerland. Springer.

Matej Ulčar and Marko Robnik-Šikonja. 2021. Training dataset and dictionary sizes matter in bert models: the case of baltic languages.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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.

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*. Association for Computational Linguistics.