Multilingual coreference resolution: Adapt and Generate

Tatiana Anikina*, **Natalia Skachkova***, **Anna Mokhova**
DFKI / Saarland Informatics Campus, Saarbrücken, Germany
tatiana.anikina@dfki.de; natalia.skachkova@dfki.de
annmo00006@stud.uni-saarland.de

Abstract

The paper presents two multilingual coreference resolution systems submitted for the CRAC Shared Task 2023. The *DFKI-Adapt* system achieves 61.86 F1 score on the shared task test data, outperforming the official baseline by 4.9 F1 points. This system uses a combination of different features and training settings, including character embeddings, adapter modules, joint pre-training and loss-based re-training. We provide evaluation for each of the settings on 12 different datasets and compare the results. The other submission *DFKI-MPrompt* uses a novel approach that involves prompting for mention generation. Although the scores achieved by this model are lower compared to the baseline, the method shows a new way of approaching the coreference task and provides good results with just five epochs of training.

1 Introduction

Coreference resolution is a task of finding all mentions referring to the same physical or abstract entity in the given piece of text. E.g., in sentences *"I've never been to London before. But I heard it is a lovely place"* the words *London* and *it* both refer to the real-world entity *the city of London*, and are called an antecedent and an anaphor respectively. Coreference resolution includes two sub-tasks that can be done either in a pipeline manner, or jointly: mention detection and mention clustering. They are quite challenging: (i) antecedents can be split; (ii) mentions can be discontinuous; (iii) one needs to consider the semantics of the context; (iv) there are long-distance coreference relations, etc. Coreference resolution contributes to the correct automatic text understanding, and is important for many NLP tasks, including text summarization and paraphrasing, information extraction, machine translation, question answering, etc.

The CRAC-2023 shared task (Žabokrtský et al., 2023) focuses on multilingual coreference resolution. However, the majority of language models are still being created for English, e.g., about 70% of the oral papers at ACL 2021 presented models evaluated only on English (Ruder et al., 2022). The problem is that many languages, even some of the big ones, do not have enough labeled training data, especially for specific tasks. Another issue is that training a separate model for each separate language when the task stays the same can be too time- and resource-consuming, especially when the model is large. A typical solution to this is transfer learning, when a model trained on some language(s) or task(s) is adapted to work for another one. In this paper we present our approach to transfer learning for multilingual coreference resolution.

Our first submission *DFKI-Adapt* presents a novel approach which combines joint pre-training, combined datasets for related languages, loss-based re-training, character embeddings and adapters. Our second submission *DFKI-MPrompt* integrates prompting. Prompting is a way of eliciting the desired output from a large language model (LLM). It was first introduced by Brown et al. (2020). The main motivation behind prompting is to avoid computationally expensive fine-tuning of LLMs, as they contain billions of parameters. Moreover, such models already incorporate lots of various knowledge, therefore we can simply add demonstrations to our input to help the model "understand" what we want and produce the desired output.

To summarize, our contributions are as follows.

- We investigate how to combine the existing data, features and fine-tuning approaches to improve the baseline results without larger models or additional data.
- We check if knowledge accumulated in large multilingual language models can be extracted

*Equal contribution

Proceedings of the CRAC 2023 Shared Task on Multilingual Coreference Resolution, pages 19–33
December 7, 2023

using prompt fine-tuning to perform mention detection, and if this method can compete with the state-of-the-art one.

- Some of the approaches we try have never been used for the given task before, and can be of interest for the community.

2 Related work

In this section we outline the main achievements in the area of multilingual coreference resolution, and present the approaches that are similar to our work.

Most progress in the area of multilingual coreference resolution was made due to the introduction of shared tasks. SemEval-2010 Task 1 (Recasens et al., 2010) was designed to evaluate and compare methods of coreference resolution in six languages (Catalan, Dutch, English, German, Italian, and Spanish) and used four different metrics: MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), CEAF (Luo, 2005) and BLANC (Recasens and Hovy, 2011). There were six systems submitted for this task, all of them rely on feature extraction and machine learning algorithms, like maximum entropy, decision trees, support vector machines (SVM), etc. Only two systems, UBIU (Zhekova and Kübler, 2010) and SUCRE (Kobdani and Schütze, 2010) work for all the languages.

CoNLL-2012 (Pradhan et al., 2012) was dedicated to predicting coreference in the OntoNotes corpus (Pradhan et al., 2007) which includes data in English, Chinese, and Arabic. The evaluation metrics included metrics used for SemEval-2010 and a CoNLL score representing an unweighted average of MUC, B^3 and entity based CEAF. There were 16 systems submitted for CoNLL-2012. The majority of them combine machine learning approaches mentioned earlier with the rule-based ones. The latter are typically used for mention detection. The best performing systems also heavily rely on feature engineering. As far as we can judge, most of the systems assume training a separate model for each language.

In contrast to the previous shared tasks, CRAC-2022 (Žabokrtský et al., 2022) offered much more data in different languages. The CorefUD 1.0 collection (Nedoluzhko et al., 2022) included 13 datasets in Czech, English, Polish, French, Russian, German, Catalan, Spanish, Lithuanian and Hungarian which were harmonized to the same annotation scheme and data format. The primary evaluation metric was the CoNLL score. The organizers offered a strong Transformer-based baseline (Pražák et al., 2021), which was also used for the current shared task. There were eight systems submitted.The absolute majority use deep learning approaches and rely on large pre-trained models. Importantly, most of the systems present cross-lingual models trained on all the multilingual data.

It is actually difficult to compare all these models in terms of numbers and judge how much progress has been made since SemEval-2010 for multilingual coreference resolution. First, the models were trained on quite different data. Second, despite the unification of the annotations, the definition of a mention varies across the datasets. Third, the evaluation criteria are also different, in the first place for mention boundaries detection.

Our *DFKI-Adapt* system uses a combination of different settings that includes pre-trained adapters. As far as we know, adapters (Houlsby et al., 2019; Rebuffi et al., 2017) have not been well researched for multilingual coreference resolution. Adapters represent a small amount of additional parameters that can be added as trainable task-specific weights at each layer of the transformer architecture (Vaswani et al., 2017). They have been successful on a variety tasks including speech recognition (Hou et al., 2021), cross-lingual transfer (Parovic et al., 2022) and classification tasks (Lee et al., 2022; Anikina, 2023; Metheniti et al., 2023) but there is very little research on using adapters for coreference resolution and the only work that we are aware of uses parallel data for training (Tang and Hardmeier, 2023).

The idea of prompting LLMs for the task of coreference resolution is relatively new. There are not so many papers on this topic. E.g., Perez et al. (2021) do few-shot prompting to resolve anaphora that requires commonsense knowledge using the Winograd Schema Challenge (WSC) corpus (Levesque et al., 2012). Min et al. (2022) perform similar experiments on the WSC and WinoGrande (Sakaguchi et al., 2021) data, and Yang et al. (2022) - on ECB+ (Cybulska and Vossen, 2014). Le et al. (2022) and Agrawal et al. (2022) try prompting for coreference resolution in scientific protocols and medical domain, respectively. Lin et al. (2022) experiment with few- and zero-shot anaphora resolution in the multilingual XWinograd corpus (Tikhonov and Ryabinin, 2021). In contrast to our approach, all these models do not

perform prompt fine-tuning, instead they typically include a few demonstrations into their prompts (therefore few-shot) and use much larger models, like XGLM (Lin et al., 2022), GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020) or InstructGPT (Ouyang et al., 2022). Moreover, we use prompting only for mention generation. A somewhat similar approach but without prompting was presented by Kalyanpur et al. (2020), who used the *T5* model (Raffel et al., 2020) to generate semantic roles when doing frame-semantic parsing.

3 Data

All our experiments are done using an officially provided public version of the CorefUD 1.1 data, which extends CorefUD 1.0 with new datasets in Hungarian, Norwegian and Turkish. In total, this version of CorefUD consists of 17 datasets in 12 languages. The datasets vary a lot in their sizes (see Table 7 in the Appendix B). Moreover, they represent different language families and subgroups with very different grammars and vocabularies. Also, the datasets differ in how markables are defined, e.g., some datasets omit singletons, others may annotate verbal phrases, if they are antecedents of anaphoric noun phrases (Žabokrtský et al., 2022). All this makes it very challenging to build a single model working well for all the given languages.

Intuitively, the quality of mention extraction and subsequent coreference resolution depends not only on the training data size, but also on length and complexity of the sentences and mentions, the number of mentions (including nested) in a sentence, the amount of unique named entities, etc. To get an idea about difficulty of the task, we collected some basic statistical facts about the relevant data properties. This information can be found in Table 8 in the Appendix B.

4 Multilingual coreference resolution with DFKI-Adapt

Our submission *DFKI-Adapt* is based on the baseline provided by the organizers but extended in different ways to accommodate the multi-lingual nature of the task. The *DFKI-Adapt* system integrates character embeddings, joint pre-training and fine-tuning on the datasets of the related languages. It also includes additional re-training on the documents with the higher loss and uses adapter modules that were pre-trained for each dataset.

The goal of the *DFKI-Adapt* submission is to demonstrate how one could get a substantial improvement over the baseline (+4.9 F1 points on the test and +9.07 F1 on the development partitions) without any additional data or larger models, just by leveraging the existing annotations. All experiments are performed with standard multilingual BERT and the official CRAC data. The following sections introduce our baselines, the experiments with individual settings and the final results achieved by *DFKI-Adapt*. Since the test data are not publicly available our evaluation is performed on the CRAC development set. The evaluation results on 12 datasets for different languages[1] are summarized in Table 1. The more detailed analysis with different coreference evaluation metrics is reported in Tables 3-6 in the Appendix A.

4.1 Baselines

We consider three different baselines for our system. Firstly, we use the official baseline of the shared task which was published by the organizers (*CRAC-baseline*). Secondly, we train a single joint coreference model based on multilingual BERT and use it to predict coreference chains for each dataset (*mbert-joined*). Thirdly, we train a separate model for each language and dataset present in the shared task (*mbert-separate*). The results in Table 1 demonstrate that *mbert-joined* consistently outperforms *mbert-separate* indicating that joint training on the combination of all datasets is a good strategy for coreference resolution. The main baseline to which we compare different settings in the following sections is the official *CRAC-baseline*.

4.2 Adapters

We add adapters to multilingual BERT and then fine-tune them for each dataset separately. Then we load the pre-trained adapters and train a new coreference resolution model for each dataset from scratch but with the pre-trained adapter weights. In one setting, *task-adapters-frozen*, we do not further train the adapters, while the rest of the model is being tuned on the coreference resolution task. In another setting, *task-adapters-tuned*, we continue training the adapters together with the rest of the model. According to the experimental results

[1]For some languages several datasets were available and we selected a single dataset for each language as a representative. Although the differences between the datasets can also occur within a single language, we evaluated one dataset per language given the limited time, resources and the goal of comparing different languages rather than the datasets. Further details can be found in the Appendix A.

Dataset	mbert-joined	mbert-separate	char-embed	joined-pre-training	combined-datasets	loss-re-training	task-adapters-frozen	task-adapters-tuned	DFKI-Adapt	CRAC-baseline
ca_ancora	**68.97**	65.06	66.56	68.72	66.29	65.59	66.19	61.99	68.34	65.60
cs_pdt	66.35	65.30	67.45	68.32	66.62	65.36	66.35	61.18	**68.60**	65.66
en_gum	65.80	52.01	54.05	62.41	35.25	51.38	51.49	47.54	**69.63**	66.87
fr_democrat	59.74	58.85	58.88	60.97	61.09	57.81	57.88	52.50	**62.34**	57.22
de_potsdamcc	65.77	58.92	55.16	62.03	67.12	59.77	64.28	60.27	**69.29**	56.07
hu_szegedkoref	59.78	59.98	59.53	62.29	60.42	60.13	57.39	53.70	**62.60**	58.96
lt_lcc	71.22	69.09	69.55	73.18	**75.76**	69.47	68.05	64.95	73.08	66.96
no_bokmaal	69.81	68.47	69.11	72.26	69.09	67.65	68.83	64.53	**72.45**	58.44
pl_pcc	65.41	63.64	65.32	**66.38**	66.21	63.74	64.30	59.44	65.89	64.17
ru_rucor	62.08	62.11	63.84	66.54	64.58	63.26	61.73	57.97	**67.50**	63.04
es_ancora	67.00	66.37	67.99	69.82	66.64	66.29	66.99	62.53	**70.07**	67.00
tr_itcc	31.66	31.35	17.98	30.80	33.88	23.28	20.68	6.91	**37.80**	16.15

Table 1: CoNLL F1 scores on the development data. The best performing setting is in bold

shown in Table 1, for *task-adapters-frozen* the results differ significantly between the datasets. E.g., we can see that the model trained on the German data gives an improvement of +8.21 F1 points compared to the *CRAC-baseline* and for Turkish the improvement is +4.53 F1 points. Polish and Czech also have small gains in performance when using pre-trained adapters (+0.13 and +0.69 F1 points correspondingly). However, Hungarian has a drop of -1.57 F1 points compared to the CRAC baseline.

We also observe that using pre-trained adapters and then freezing them consistently outperforms the version with tunable adapters. Compared to the CRAC baseline the latter model underperforms by 4.39 F1 points on average. We notice that using language-specific pre-trained adapters gives model a "warm start" and it starts with a slightly better performance, e.g., the ratio of the correctly predicted to gold mention spans is higher than if we start training the model from scratch, without any pre-trained adapters.

4.3 Character embeddings

For character embeddings we consider 273 characters which include the alphabet letters of all relevant languages plus some additional symbols such as currency or copyright signs. A symbol has to occur more than 5 times in the training set in order to be included in our list of the frequent characters. After making the character list we run bi-LSTM to encode every token in the data.

Then in the coreference resolution model we add an extra layer that projects character embeddings from 300 to 100 dimensions and concatenate the character embeddings of the start and the end of each span with the corresponding BERT embeddings. We observe that adding character embeddings gives a small boost in performance compared to the CRAC baseline (+0.77 F1 points on average). Interestingly, the only two languages which show a decrease in performance are German and English, all other languages show some improvement and the largest gains are attributed to Norwegian +10.67 F1 and Lithuanian +2.59 F1.

4.4 Joined pre-training

As discussed in Section 3, the available datasets are quite different. However, since in all the cases the task is to identify and cluster coreferent mentions we believe that patterns relevant for coreference resolution in one language may prove to be helpful for another. Hence, we pre-train one multilingual BERT model on all datasets combined together and then we continue fine-tuning this model on each language separately. We restrict the number of the pre-training steps to 100,000 and leave all other hyper-parameters unchanged. This setting with the joined pre-training is beneficial for all languages and it brings an average improvement of +4.8 F1 points on the development data compared to the CRAC baseline.

4.5 Combined datasets

In the *combined-datasets* setting we test whether combining the training sets of the related languages can boost the performance. E.g., for Spanish we combine it with the training sets for other Romance languages that include Catalan and French, and for Czech we combine both datasets for this language (*cs_pdt* and *cs_pcedt*) together with the annotations for Polish and Russian. Note that we do not adjust for any differences in the dataset size and do not

balance the amount of samples that might have negatively impacted the performance in some of the cases (e.g., Spanish and Catalan data have more than 1,000 documents each, whereas French has only 50 documents).

The results in Table 1 show that combining the datasets of the related languages is a good approach in many cases, although it seems to help some languages more than the others (e.g., it brings +11.05 F1 points for *de_potsdamcc* but only +0.96 F1 for *cs_pdt*). We notice that this method is especially beneficial for those cases where we have a relatively small number of annotated documents (e.g., French with only 50 documents in the training set and Lithuanian with 80). Also, perhaps due to the differences in the annotation format, for some languages we notice a significant drop in performance when we train on the combined datasets. E.g., the model trained on the *en_gum* data combined with *en_parcorfull*, *de_potsdamcc* and *de_parcorfull* datasets shows poor performance in our experiments, achieving only 35.25 F1. Further ablation studies and error analysis are needed to find the exact cause of this issue.

In some cases finding datasets in related languages is not possible and we combine the corpora based on other linguistic similarities, e.g., both Hungarian and Turkish are agglutinative languages and both of them benefit from the combined datasets (see Table 1).

4.6 Loss-based re-training

In the *loss-re-training* setting we store the loss associated with each document per epoch and at the end of each epoch we sort the documents by their corresponding losses and take the 10% of the most difficult documents (i.e., the ones with the highest loss) to continue additional training. This means that we effectively fine-tune our models on particularly difficult instances.

This approach brings an average improvement of +0.63% F1 points across all datasets, as shown in Table 1, but the gains differ between the languages. E.g., the *lt_lcc* and *tr_itcc* data show substantial improvements with the loss-based re-training: +2.51 and +7.13 F1 points respectively. However, some datasets (e.g., *es_ancora*, *cs_pdt* and *en_gum*) show worse performance.

The discrepancy is potentially caused by the imbalance in the amount of the available training data between the datasets. The datasets with the fewer documents (e.g., *tr_itcc* with 19 and *lt_lcc* with 80) seem to benefit from the loss-based re-training while other datasets with relatively large amount of documents do not benefit from it (e.g., *es_ancora* with 1,080 documents or *cs_pdt* with 2,533). In the future we would like to explore this fine-tuning approach in more detail and apply it to different low-resource settings using various metrics to order and select difficult documents (e.g., ordering them by entropy or surprisal).

4.7 DFKI-Adapt

Our submission *DFKI-Adapt* combines the best-performing configurations as described above. It includes *joined-pre-training* for 100,000 steps together with the *combined-datasets* setting for fine-tuning on the combined training data for the related languages. It also integrates character embeddings as in the *char-embedding* configuration. Additionally, we fine-tune each model on the 10% of the most difficult documents per dataset (as in *loss-re-training*) and we also include the pre-trained adapter modules as in *task-adapters-frozen*.

DFKI-Adapt consistently outperforms all three baselines (*mbert-joined*, *mbert-separate* and *CRAC-baseline*) and for most of the languages it gives the best performance on the development set, although for some datasets (e.g., *pl_pcc* and *lt_lcc*) other configurations such as *joined-pre-training* or *combined-datasets* perform slightly better than *DFKI-Adapt* (see Table 1 for comparison). On the official test set our *DFKI-Adapt* system achieves 61.86 CoNLL F1 score (+4.89 F1 points compared to the CRAC baseline) and on the development set it achieves 68.06 CoNLL F1 score (+9.07 F1 points compared to the baseline).

All our models are trained on either NVIDIA RTX A6000 with 48 GB memory or NVIDIA A100-SXM4 with 40 GB memory. We use the hyper-parameter settings as defined in the baseline configuration file[2] and train the models for the same amount of epochs. For the models that use adapters we set the BERT learning rate to 1e-05 and the task learning rate to 2e-4. We set the dropout rate to 0.4 and the mention loss coefficient to 0. For optimizing the network we employ *AdamW* and a linear schedule with warm-up.

[2]See https://github.com/ondfa/coref-multiling/ for the configuration details and the hyper-parameter settings.

5 Multilingual coreference resolution with DFKI-MPrompt

In this section we first present our approach to mention identification as generation, then explain how we adapt the baseline to work with the mentions generated by our model. We discuss the results obtained by our system, analyse the mistakes and outline possible improvements.

5.1 Mention generation

The absolute majority of modern coreference resolution models, including the baseline provided for this shared task, use span ranking with pruning to identify mentions. As pointed out in Section 3, the results depend on many factors, such as how the markables are defined in the dataset, the dataset size, domain and language, etc. E.g., the baseline[3] reaches up to 85.16 F1 in mention identification on the *no_nynorsk* and only about 54.65 F1 on the *tr_itcc* development data. Correct mention identification is crucial for successful coreference resolution. The same baseline achieves the F1 score of only 38.17 in coreference resolution on the *de_parcorfull* development data, if it has to predict the mentions. However, if the gold mentions are given, the F1 score reaches 91.90 points on the same data.

Motivated by the recent success of prompting LLMs for various downstream NLP tasks, we decide to try casting mention identification problem as a generation task using a simple prefix prompt. Theoretically, mention generation offers certain advantages in comparison with the span-ranking approach, e.g., (a) no pruning is required; (b) it is possible to generate discontinuous and nested mentions; (c) both input and output are in natural language and therefore are easy to analyze for a human. Moreover, as far as we are aware, no one has tried mention generation as a way to identify mentions for coreference resolution before.

We use a family of multilingual *T5* models (Xue et al., 2021), namely *mT5-base* and *mT5-large* with 580M and 1.2B parameters, respectively. We omit the demonstrations in our prompt, as they can make the input quite lengthy, and are unlikely to work with relatively small models. Instead, we use a prefix consisting of five tunable embeddings prepended to our input. This method was first presented by Li and Liang (2021). For all our experiments we employ the Openprompt library (Ding et al., 2022), which we locally extend so that it works with multilingual *T5* models.

Our task is formulated as follows. Given an input string consisting of one sentence, the desired output should include all mentions contained in the given sentence together with their start and end indices in brackets. Generated mentions should be separated from each other with a delimiter (a vertical bar). To help the model generate indices, we modify the input by adding the corresponding index to each token, like Kalyanpur et al. (2020) do. Example 5.1 shows the approach. Both the model and the input embeddings stay frozen, and only prefix embeddings, which are added to the input under the hood, get updated during the prompt training. The prompt itself is given in Example 5.2.

Example 5.1. Model input and output
Input: *0 já 1 Jsem 2 prý 3 v 4 USA 5 a 6 hry 7 skončily 8 , 9 uvedl 10 de 11 Merode 12 .*
Output: *já (0-0) | de Merode (10-11) | hry (6-6) | v USA (3-4)*

Example 5.2. Prompt
0 já 1 Jsem 2 prý 3 v 4 USA 5 a 6 hry 7 skončily 8 , 9 uvedl 10 de 11 Merode 12 . Find all valid mentions: [MASK]

The total number of training and development examples makes up 178,028 and 24,404 sentences, respectively. Sentences without mentions are omitted. As discontinuous mentions represent only a tiny portion of all the mentions, we omit them as well. We set the maximum input length to 256 tokens, and expect the generated output also to be no longer than that. The training is done on one NVIDIA GeForce GTX TITAN X GPU with 12 GB memory on all the available multilingual training data for five epochs with the batch size 1, the *AdamW* optimizer, learning rate of 5e-5 and a linear schedule with warm-up. It takes about a week to complete the training.

As we do not have access to the gold test data, we evaluate our mention generation approach on the development partition. The results in terms of recall, precision and F1 are presented in Table 2. The table also includes mention detection scores achieved by the baseline. We see that the baseline results are more than +10 points higher on the combined data, with our approach showing better F1 only for the *de_parcorfull* and *tr_itcc* corpora. However, baseline scores are not directly compa-

[3]We consider the version trained on all the available multilingual train data in CorefUD 1.1 with singletons excluded from evaluation.

rable with the scores reached using the prompting approach. To calculate the baseline's scores we use the predicted clusters with all singleton clusters removed. To be fair, we also exclude all gold singleton clusters from the evaluation. In contrast to that, our mention generation is done before coreference resolution. Thus, it is impossible to remove any singletons, as no clusters exist yet.

Table 2 shows that our method allows to get decent results, with *mT5-large* typically producing much better scores than *mT5-base*. As expected, better scores are normally achieved for larger datasets. However, there are some exceptions, e.g., the F1 score is 72.92 for *de_potsdamcc* and only 62.46 for *cs_pdt*, which have 4,061 and 142,951 continuous mentions in the training data, respectively. This points at the fact that some datasets contain "easier" mentions than others. Interestingly, the precision is always higher than recall, except for two *parcorfull* corpora. This may be an indication that the definition of a mention used to annotate them differs a lot from those applied to other datasets.

To better understand the results and find some possible space for improvement, we analysed the mistakes made by our approach. First, as expected, we discovered that shorter mentions in shorter sentences are more likely to be generated correctly - the average length of correctly generated and missing (not generated) mentions makes 2.03 and 5.86 tokens, respectively. The average length of sentences in which all mentions were identified correctly is about 11.67 tokens, while the sentences in which at least one mention was generated incorrectly (either a mention itself, or its indices, or both) contain 23.41 tokens on average. Second, among 21,133 wrong outputs (a) 379 (1.79%) do not have brackets with indices, and only four instances among them are correct mention strings; (b) 752 (3.56%) have a wrong delimiter, thus representing merged outputs, of which only 29 are fully correct, five are correct but have wrong indices, and 544 are wrong mentions with correct indices. Example B.1 in the Appendix B illustrates the problem. As for the rest 20,002 wrong outputs (i.e. cases consisting of one mention and one index pair), we found out that 245 (1.22%) of them have wrong indices, and 5,690 (28.45%) - wrong mention strings. Other 14,067 (70.33%) outputs have both wrong mentions and wrong indices. Finally, we detected that the average length of outputs with correct indices but wrong strings varies from 10.85 to 13.03 tokens, which shows that the model is still capable to deal with longer mentions. More information on that can be found in Appendix B.

Based on the error analysis, we would suggest the following modifications of the approach. First, simplification of the desired output seems to be promising. Our current output pattern is quite challenging, instead, we can ask the model to produce only the indices of mentions, like *'10-11'*, or a direct substring of the input string, like *'10 de 11 Merode'*. This would probably help to deal with missing spaces before punctuation marks, which make a large part of all mistakes. Next, we believe that training the prompt for more epochs, as well as tuning some other hyperparameters, like the number of prefix tokens, may lead to performance improvements. Experiments with other types of templates and a better prompt engineering may also be beneficial. Finally, it is possible to group the datasets depending on the mention definitions, train several prompts, and do prompt ensembling.

5.2 Coreference resolution

As we have a separate module to identify the mentions, we slightly change the baseline so that it performs only coreference resolution. This means that the model does not need to create spans, assign scores to them and do the pruning, because the mentions are already known. We re-train the baseline on gold mentions (including singletons) with all the default hyperparameters, and then evaluate it on mentions generated with our prefix prompt. While the original baseline reaches 66.78 CoNLL score on the combined development data, adding our prompt-based module to it causes about -7 points drop in performance. This is not unexpected, as mention identification results achieved by our method were in general worse than those produced by the baseline. Only for three datasets the CoNLL scores were higher, and for two out of these three our approach also demonstrated better mention identification results in comparison with the baseline. All scores can be found in Table 9 in the Appendix B. On average, according to the official leaderboard, our model reaches the CoNLL scores of 57.22 and 53.76 on the development and test data, respectively. In both cases it takes the last place on the list of eight (development) and ten (test) submissions. Still, we find the scores decent, considering how little effort our method takes.

Data	num men	mT5-base			mT5-large			baseline
		R	P	F1	R	P	F1	F1
all	108,006	55.42	72.56	62.84	63.53	75.80	69.13	**79.90**
ca_ancora	7,280	46.11	67.48	54.79	54.52	71.23	61.77	**81.55**
cs_pcedt	23,784	56.90	67.16	61.61	63.23	71.13	66.95	**80.90**
cs_pdt	20,955	47.79	71.34	57.24	54.12	73.83	62.46	**78.76**
en_gum	5,508	61.86	80.54	69.97	71.10	81.98	76.15	**80.24**
en_parcorfull	79	69.62	27.36	39.29	70.89	25.34	37.33	**58.13**
fr_democrat	7,032	61.52	78.21	68.87	72.16	80.01	75.88	**78.63**
de_parcorfull	93	65.59	44.20	52.81	72.04	44.67	**55.14**	53.89
de_potsdamcc	558	56.99	70.20	62.91	69.00	77.31	72.92	**73.47**
hu_korkor	448	47.54	66.15	55.32	54.91	68.72	61.04	**70.85**
hu_szegedkoref	1,458	54.87	61.73	58.10	60.84	66.10	63.36	**68.23**
lt_lcc	366	48.09	55.17	53.39	55.46	63.04	59.01	**77.06**
no_bokmaal	6,446	65.54	80.80	72.38	76.79	85.23	80.79	**84.07**
no_nynorsk	5,193	67.24	79.76	72.97	77.89	83.83	80.75	**85.16**
pl_pcc	18,857	56.77	75.88	64.95	66.27	79.03	72.09	**77.49**
ru_rucor	2,297	68.35	78.70	73.16	75.49	80.61	77.97	**83.43**
es_ancora	7,161	46.64	66.91	54.97	54.11	71.83	61.72	**82.56**
tr_itcc	491	58.45	75.13	65.75	65.38	76.98	**70.70**	54.65

Table 2: Mention identification results. *All* stands for all the development data taken together (not the average).

6 Conclusion

In this paper we presented our systems for multilingual coreference resolution.

Our *DFKI-Adapt* submission leverages the existing data in different ways including joint pre-training, integrating adapters, adding character embeddings and loss-based re-training. It achieves 61.86 F1 on the official test set and 68.06 F1 on the development set. We provide a comparison of different settings for 12 languages from the CRAC shared task. Based on our analysis, joined pre-training with further fine-tuning on the respective dataset is the most beneficial setting per se but the largest gains can be achieved with the combination of different settings as implemented in the *DFKI-Adapt* system. Our experiments also show that while injecting the pre-trained adapter weights can be helpful for many languages, these pre-trained weights should not be further updated during training. In the future we would like to experiment more with the language-specific vs. task-specific adapters and test whether cross-lingual transfer via adapters could further improve the performance on the coreference resolution task.

Our second submission *DFKI-MPrompt* relies on a novel prompt-based approach for mention identification. It generates all possible mention strings together with their indices, given a sentence. Although the obtained scores were lower than baseline scores for the majority of the datasets, our method still has some potential. First, it can be improved by applying a better template, more optimal hyperparameters and a larger model. Second, it could be used as an additional tool helping span-based mention-ranking state-of-the-art models find mentions that are especially challenging for them, like split antecedents or discontinuous mentions. As a possible next step we plan experiments to check if our approach is capable of such a task.

Limitations

We believe that our *DFKI-Adapt* system could be further improved by adding more adapter weights and experimenting with the cross-lingual transfer learning. The current system uses adapters as a way of additional pre-training of the encoder but it would be interesting to see whether adapters for different languages can also benefit each other, similarly to the *combined_datasets* setting.

Casting mention identification as a prompt-based generation task also has its limitations. Using prompting, good results (sometimes even better than state-of-the-art) can be typically obtained with very large models that are not always freely available and require lots of computational resources. Even with relatively small models, like *T5*, prompt-tuning/inference may take several days, if one does not have access to powerful GPUs. This makes the process of finding the optimal prompt and hyperparameters very time-consuming.

Acknowledgements

The authors are supported by the German Ministry of Education and Research (BMBF): T. Anikina in project CORA4NLP (grant Nr. 01IW20010); N. Skachkova, A. Mokhova in IMPRESS (grant Nr. 01IS20076).

References

Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. 2022. Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1998–2022, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tatiana Anikina. 2023. Towards efficient dialogue processing in the emergency response domain. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 212–225, Toronto, Canada. Association for Computational Linguistics.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, volume 1, pages 563–566. Citeseer.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4545–4552, Reykjavik, Iceland. European Language Resources Association (ELRA).

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Haitao Zheng, and Maosong Sun. 2022. OpenPrompt: An open-source framework for prompt-learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 105–113, Dublin, Ireland. Association for Computational Linguistics.

Wenxin Hou, Hanlin Zhu, Yidong Wang, Jindong Wang, Tao Qin, Renjun Xu, and Takahiro Shinozaki. 2021. Exploiting adapters for cross-lingual low-resource speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:317–329.

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 *Proceedings of International Conference on Machine Learning*, pages 2790–2799. PMLR.

Aditya Kalyanpur, Or Biran, Tom Breloff, Jennifer Chu-Carroll, Ariel Diertani, Owen Rambow, and Mark Sammons. 2020. Open-domain frame semantic parsing using Transformers.

Hamidreza Kobdani and Hinrich Schütze. 2010. Sucre: A modular system for coreference resolution. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 92–95.

Nghia T. Le, Fan Bai, and Alan Ritter. 2022. Few-shot anaphora resolution in scientific protocols via mixtures of in-context experts. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*.

Jaeseong Lee, Seung-won Hwang, and Taesup Kim. 2022. FAD-X: Fusing adapters for cross-lingual transfer to low-resource languages. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 57–64, Online only. Association for Computational Linguistics.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd schema challenge. In *Proceedings of the Thirteenth international conference on the principles of knowledge representation and reasoning*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32.

Eleni Metheniti, Chloé Braud, Philippe Muller, and Laura Rivière. 2023. DisCut and DiscReT: MELODI at DISRPT 2023. In *Proceedings of the 3rd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2023)*, pages 29–42, Toronto, Canada. The Association for Computational Linguistics.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.

Anna Nedoluzhko, Michal Novák, Martin Popel, Zdeněk Žabokrtský, Amir Zeldes, and Daniel Zeman. 2022. CorefUD 1.0: Coreference meets Universal Dependencies. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4859–4872.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Marinela Parovic, Goran Glavas, Ivan Vulic, and Anna Korhonen. 2022. BAD-X: Bilingual adapters improve zero-shot cross-lingual transfer. In *Proceedings of North American Chapter of the Association for Computational Linguistics*.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *Advances in neural information processing systems*, 34:11054–11070.

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

Sameer S. Pradhan, Eduard Hovy, Mitch Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2007. OntoNotes: A unified relational semantic representation. In *Proceedings of International Conference on Semantic Computing (ICSC 2007)*, pages 517–526. IEEE.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Marta Recasens and Eduard Hovy. 2011. BLANC: Implementing the rand index for coreference evaluation. *Natural language engineering*, 17(4):485–510.

Marta Recasens, Lluís Màrquez, Emili Sapena, M Antònia Martí, Mariona Taulé, Véronique Hoste, Massimo Poesio, and Yannick Versley. 2010. SemEval-2010 task 1: Coreference resolution in multiple languages. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 1–8.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2022. Square one bias in NLP: Towards a multi-dimensional exploration of the research manifold. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2340–2354, Dublin, Ireland. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Gongbo Tang and Christian Hardmeier. 2023. Parallel data helps neural entity coreference resolution. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3162–3171, Toronto, Canada. Association for Computational Linguistics.

Alexey Tikhonov and Max Ryabinin. 2021. It's All in the Heads: Using Attention Heads as a Baseline for Cross-Lingual Transfer in Commonsense Reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3534–3546, Online. Association for Computational Linguistics.

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.

Marc Vilain, John D Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

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.

Xiaohan Yang, Eduardo Peynetti, Vasco Meerman, and Chris Tanner. 2022. What GPT knows about who is who. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 75–81, Dublin, Ireland. Association for Computational Linguistics.

Zdeněk Žabokrtský, Miloslav Konopík, Anna Nedoluzhko, Michal Novák, Maciej Ogrodniczuk, Martin Popel, Ondřej Pražák, Jakub Sido, and Daniel Zeman. 2023. Findings of the second shared task on multilingual coreference resolution. In *Proceedings of the Sixth Workshop on Computational Models of Reference, Anaphora and Coreference (CRAC 2023)*, pages 1–18.

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.

Desislava Zhekova and Sandra Kübler. 2010. UBIU: A language-independent system for coreference resolution. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 96–99.

A DFKI-Adapt vs. other configurations

Tables 3-6 present the evaluation results on the development data for 12 languages. The CoNLL score is compared with the scores achieved by separate metrics, namely MUC, B^3 and CEAFE.

B Mention generation

B.1 Data statistics

Tables 7 and 8 present the main statistical facts about the CorefUD 1.1 data that help explain mention identification results. Table 7 illustrates the differences between the datasets in terms of size by providing information about the number of documents, sentences and tokens in the training and development partitions of separate corpora.

Table 8 gives for each dataset information about sentence lengths, number of continuous and discontinuous mentions, average number of mentions in a sentence, and average mention lengths. We see that the sentences in CorefUD 1.1 may be of different length and contain different number of mentions. On average, a sentence consists of 21 tokens, the shortest sentences (14.93 tokens on average) can be found in the *tr_itcc* dataset, and the longest (34.06 tokens on average) - in *es_ancora*. The total number of continuous and discontinuous mentions in all the training data makes up 794,643 and 5,543, respectively. Typically, a sentence includes 4.46 mentions, and the number of mentions in a sentence correlates with its length, e.g., in *es_ancora* a sentence contains 5.32 mentions on average, and in *tr_itcc* - only 1.80 mentions. Some sentences do not contain any mentions. Normally, a mention consists of 3.32 tokens, the longest mentions (4.98 tokens on average) occur in *es_ancora*, the shortest (1.53 tokens on average) - in the *lt_lcc* dataset.

B.2 Coreference resolution results

Table 9 presents coreference resolution results for all the development partitions of separate datasets. Note that we also evaluate our approach on the combined data (*all*). In contrast to this, the official leaderboard shows the averaged score based on separate results. Table 9 shows precision, recall and F1 (CoNLL) score produced by two versions of the baseline model. The *predicted mentions* section contains the results achieved by the original baseline trained on all the available training data. The *gold mentions* part - the points produced by baseline trained on all the gold mentions, given gold development mentions. Finally, the *generated mentions* section shows the scores reached by the baseline trained on the gold mentions when evaluated on the mentions generated by our mention identification module.

B.3 Mention generation errors

Example B.1 shows a typical error case. First, the generated mentions can not be separated, because the delimiter "|," is wrong. Second, one of the two gold mentions, namely *", fundador de la aerolínea Spantax"* starts with a comma, which the model fails to generate. However, despite the missing comma, the indices *(4-9)* corresponding to this mention are generated correctly.

Example B.1. Generated merged output
'Rodolfo Bay Wright, fundador de la aerolínea Spantax (1-9) |, fundador de la aerolínea Spantax (4-9)'
Gold output
'Rodolfo Bay Wright, fundador de la aerolínea Spantax (1-9) | , fundador de la aerolínea Spantax (4-9)'

We additionally analysed cases where the generated mention strings were wrong but the indices correct. It turned out that *mT5* tends to skip spaces before punctuation marks, while gold mentions have them, e.g., the model generates *'Eugene, Oregon'* instead of *'Eugene , Oregon'*. Moreover, we found out that many mentions in the gold data may start and/or end with a comma, like *', Juan José Ibarretxe ,'*, which was obviously confusing for the model.

Setting	es_ancora				de_potsdamcc				tr_itcc			
	MUC	B^3	$CEAFE$	$CoNLL$	MUC	B^3	$CEAFE$	$CoNLL$	MUC	B^3	$CEAFE$	$CoNLL$
mbert-joined	74.65	67.03	66.20	67.00	69.06	64.22	64.02	65.77	41.91	23.76	29.32	31.66
mbert-separate	71.87	64.14	63.09	66.37	63.52	58.03	55.22	58.92	41.48	22.64	29.94	31.35
char-embedding	73.42	66.03	64.52	67.99	61.98	54.88	48.63	55.16	26.89	10.83	16.20	17.98
joined-pre-training	74.97	68.00	66.50	**69.82**	66.67	59.99	59.44	62.03	43.30	21.06	28.03	30.80
combined-datasets	72.37	64.69	62.85	66.64	71.72	65.67	63.99	**67.12**	45.14	25.23	31.26	**33.88**
loss-re-training	71.87	64.34	62.67	66.29	64.98	58.03	56.28	59.77	36.28	15.87	17.68	23.28
task-adapters-frozen	72.67	64.90	63.38	66.99	67.08	62.14	63.63	64.28	30.47	12.38	19.20	20.68
task-adapters-tuned	68.69	60.28	58.61	62.53	65.02	57.45	58.33	60.27	7.51	3.79	9.45	6.91
DFKI-Adapt	75.26	68.21	66.74	**70.07**	72.99	67.46	67.40	**69.29**	50.64	29.69	33.06	**37.80**
CRAC-baseline	-	-	-	67.00	-	-	-	56.07	-	-	-	16.15

Table 3: Coreference resolution results on the development data for Spanish, German and Turkish

Setting	pl_pcc				cs_pdt				hu_szegedkoref			
	MUC	B^3	$CEAFE$	$CoNLL$	MUC	B^3	$CEAFE$	$CoNLL$	MUC	B^3	$CEAFE$	$CoNLL$
mbert-joined	72.44	63.48	60.30	65.41	71.21	64.12	63.72	66.35	61.91	57.62	59.80	59.78
mbert-separate	71.38	61.60	57.93	63.64	70.76	63.50	61.63	65.30	62.56	57.81	59.56	59.98
char-embedding	72.77	63.23	59.97	65.32	72.31	65.53	64.52	67.45	61.57	57.36	59.67	59.53
joined-pre-training	73.63	64.28	61.25	**66.38**	73.11	66.59	65.26	**68.32**	64.43	60.14	62.29	**62.29**
combined-datasets	73.31	64.09	61.24	**66.21**	71.86	64.75	63.25	66.62	62.55	57.91	60.78	60.42
loss-re-training	71.33	61.64	58.26	63.74	70.80	63.50	61.78	65.36	62.43	57.79	60.17	60.13
task-adapters-frozen	71.47	61.89	59.53	64.30	71.58	64.44	63.04	66.35	59.36	54.49	58.31	57.39
task-adapters-tuned	67.75	57.12	53.45	59.44	66.76	58.92	57.86	61.18	55.37	51.15	54.59	53.70
DFKI-Adapt	73.20	63.63	60.86	65.89	73.33	66.78	65.68	**68.60**	65.49	60.37	61.95	**62.60**
CRAC-baseline	-	-	-	64.17	-	-	-	65.66	-	-	-	58.96

Table 4: Coreference resolution results on the development data for Polish, Czech and Hungarian

Setting	ca_ancora				fr_democrat				en_gum			
	MUC	B^3	$CEAFE$	$CoNLL$	MUC	B^3	$CEAFE$	$CoNLL$	MUC	B^3	$CEAFE$	$CoNLL$
mbert-joined	74.30	65.87	66.74	**68.97**	71.41	51.74	56.05	59.74	77.66	63.28	56.45	**65.80**
mbert-separate	71.20	62.06	61.93	65.06	69.95	50.11	56.50	58.85	65.91	49.89	40.22	52.01
char-embedding	72.47	63.58	63.62	66.56	69.72	50.37	56.55	58.88	67.64	52.05	42.46	54.05
joined-pre-training	74.23	65.89	66.05	**68.72**	72.06	52.32	58.54	60.97	74.72	61.68	50.82	62.41
combined-datasets	72.27	63.30	63.31	66.29	72.43	52.27	58.56	**61.09**	42.01	32.46	31.27	35.25
loss-re-training	71.63	62.48	62.66	65.59	69.60	49.37	54.45	57.81	65.56	50.11	38.47	51.38
task-adapters-frozen	71.96	63.21	63.40	66.19	69.41	48.82	55.41	57.88	65.27	49.97	39.24	51.49
task-adapters-tuned	68.70	58.70	58.57	61.99	65.17	43.29	49.04	52.50	60.51	44.99	37.12	47.54
DFKI-Adapt	74.01	65.45	65.56	68.34	72.74	54.47	59.80	**62.34**	80.43	68.38	60.08	**69.63**
CRAC-baseline	-	-	-	65.60	-	-	-	57.22	-	-	-	66.87

Table 5: Coreference resolution results on the development data for Catalan, French and English

Setting	lt_lcc				ru_rucor				no_bokmaal			
	MUC	B^3	$CEAFE$	$CoNLL$	MUC	B^3	$CEAFE$	$CoNLL$	MUC	B^3	$CEAFE$	$CoNLL$
mbert-joined	73.44	69.55	70.68	71.22	74.63	54.16	57.46	62.08	80.10	66.54	62.79	69.81
mbert-separate	69.92	66.49	70.86	69.09	73.83	55.23	57.27	62.11	77.07	67.31	61.04	68.47
char-embedding	71.08	66.91	70.66	69.55	75.24	56.65	59.64	63.84	78.00	67.33	61.99	69.11
joined-pre-training	75.49	71.88	72.17	**73.18**	77.28	59.43	62.92	**66.54**	81.32	71.09	64.36	**72.26**
combined-datasets	77.33	73.85	76.12	**75.76**	75.58	57.23	60.93	64.58	78.19	67.80	61.28	69.09
loss-re-training	71.25	67.59	69.58	69.47	74.60	55.91	59.27	63.26	76.92	66.26	59.76	67.65
task-adapters-frozen	70.97	66.82	66.37	68.05	73.52	54.46	57.20	61.73	77.81	66.85	61.83	68.83
task-adapters-tuned	66.92	62.62	65.32	64.95	69.18	51.08	53.65	57.97	74.46	62.70	56.44	64.53
DFKI-Adapt	74.79	71.39	73.06	73.08	78.77	60.32	63.40	**67.50**	81.39	70.95	65.01	**72.45**
CRAC-baseline	-	-	-	66.96	-	-	-	63.04	-	-	-	58.44

Table 6: Coreference resolution results on the development data for Lithuanian, Russian and Norwegian

Data	train			dev		
	#doc	#sent	#tok	#doc	#sent	#tok
all	9,595	194,460	3,899,182	1,325	26,698	547,869
ca_ancora	1,011	10,638	337,876	131	1,443	49,695
cs_pcedt	1,875	39,832	964,606	337	6,960	169,211
cs_pdt	2,533	38,725	670,889	316	5,228	90,645
en_gum	151	8,548	147,949	22	1,117	19,654
en_parcorfull	15	457	8,765	2	48	1,155
fr_democrat	50	10,382	228,100	46	1,192	28,279
de_parcorfull	15	457	8,649	2	48	1,098
de_potsdamcc	142	1,817	26,677	17	216	3,376
hu_korkor	76	1,086	21,063	9	130	2,715
hu_szegedkoref	320	7,138	104,428	40	846	12,355
lt_lcc	80	1,330	30,082	10	213	3,385
no_bokmaal	284	13,071	203,220	31	1,317	21,658
no_nynorsk	336	10,320	172,764	28	1,158	17,977
pl_pcc	1,463	28,722	431,999	183	3,573	53,999
ru_rucor	145	7,969	123,599	18	1,286	21,139
es_ancora	1,080	11,336	373,402	131	1,367	46,668
tr_itcc	19	3,532	45,114	2	556	4,860

Table 7: Number of documents, sentences and tokens in CorefUD 1.1

Data	Sent len			num	num	# cont in sent			men
	max	min	avg	cont	disc	max	min	avg	len
all	405	1	21.00	794,643	5,543	156	0	4.46	3.32
ca_ancora	239	2	32.61	48,705	1	27	1	4.81	4.94
cs_pcedt	134	1	25.27	138,713	1,044	22	0	3.85	3.83
cs_pdt	195	1	18.25	142,951	1,958	25	0	3.99	3.22
en_gum	110	1	18.32	41,649	0	40	1	5.21	3.05
en_parcorfull	58	4	20.09	717	5	11	0	2.13	2.02
fr_democrat	125	1	22.34	63,562	0	40	1	6.25	2.37
de_parcorfull	60	4	20.67	749	2	11	1	2.33	1.94
de_potsdamcc	54	2	16.35	4,061	265	13	0	2.72	2.76
hu_korkor	79	5	19.85	3,167	19	16	0	3.12	2.46
hu_szegedkoref	123	2	15.89	12,555	45	19	0	2.23	1.75
lt_lcc	88	2	23.56	3,723	0	15	1	3.09	1.53
no_bokmaal	88	1	15.86	61,183	339	28	0	4.80	2.94
no_nynorsk	120	1	16.97	51,450	211	34	1	5.07	3.10
pl_pcc	405	1	15.46	149,057	1,618	156	0	5.38	2.87
ru_rucor	129	1	20.24	12,576	36	34	0	2.47	1.64
es_ancora	119	2	34.06	57,223	0	23	1	5.32	4.98
tr_itcc	82	2	14.93	2,602	0	11	1	1.80	1.94

Table 8: Sentence and mention properties in training data

Data	predicted mentions			gold mentions			generated mentions		
	R	P	F1 (CoNLL)	R	P	F1 (CoNLL)	R	P	F1 (CoNLL)
all	61.14	73.73	66.78	75.05	82.22	78.42	50.56	72.76	59.58
ca_ancora	62.92	75.76	68.72	78.13	85.35	81.57	45.08	73.11	55.73
cs_pcedt	62.65	74.50	68.01	78.53	87.17	82.59	54.52	75.62	63.32
cs_pdt	58.64	75.91	66.10	74.38	81.43	77.70	44.63	74.52	55.76
en_gum	58.55	73.27	65.02	70.70	76.60	73.45	53.92	72.17	61.55
en_parcorful	65.61	38.18	48.16	90.98	91.96	91.05	65.23	35.38	45.54
fr_democrat	57.99	65.32	60.89	66.19	69.74	67.20	52.72	62.82	56.53
de_parcorfull	42.06	35.15	38.17	91.13	92.82	91.90	67.48	52.76	*58.86*
de_potsdamcc	58.81	70.72	64.16	71.65	82.51	76.56	65.02	68.25	*66.38*
hu_korkor	47.73	65.38	55.11	68.26	75.13	71.45	38.88	56.64	46.06
hu_szegedkoref	56.11	65.34	60.34	80.57	85.86	83.12	47.13	59.82	52.71
lt_lcc	65.85	80.70	72.47	89.84	92.72	91.17	55.10	73.40	62.94
no_bokmaal	66.83	76.47	71.13	69.73	77.20	72.97	60.24	74.30	66.16
no_nynorsk	67.93	75.28	71.07	70.58	76.58	73.10	63.56	74.39	68.18
pl_pcc	61.39	70.63	65.64	70.84	77.22	72.43	52.99	68.75	59.79
ru_rucor	60.91	67.73	63.26	69.07	80.78	73.81	52.18	69.33	58.92
es_ancora	62.80	77.36	69.31	76.86	86.18	81.25	46.14	76.06	57.41
tr_itcc	27.91	40.70	30.82	59.03	68.74	61.47	30.30	51.98	*36.84*

Table 9: Baseline's coreference resolution results on the development data