End-to-end Multilingual Coreference Resolution with Mention Head Prediction

Ondřej Pražák and **Miloslav Konopík**

{ondfa,konopik}@kiv.zcu.cz

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

Abstract

This paper describes our approach to the CRAC 2022 Shared Task on Multilingual Coreference Resolution. Our model is based on a state-of-the-art end-to-end coreference resolution system. Apart from joined multilingual training, we improved our results with mention head prediction. We also tried to integrate dependency information into our model. Our system ended up in 3^{rd} place. Moreover, we reached the best performance on two datasets out of 13.

1 Introduction

Coreference resolution is the task of finding language expressions that refer to the same real-world entity (antecedent) of a given text. Sometimes the corefering expressions can come from a single sentence. However, the expressions can be one or more sentences apart as well. It is necessary to see the whole document in some hard cases to judge whether two expressions are corefering adequately. This task can be divided into two subtasks. Identifying entity mentions, and grouping the mentions together according to the real-world entity they refer to. The task of coreference resolution is closely related to anaphora resolution – see (Sukthanker et al., 2020) to compare these two tasks.

This paper describes our approach to the CRAC 2022 Shared Task on Multilingual Coreference Resolution. The task is based on the CorefUD dataset (Nedoluzhko et al., 2022). The CorefUD corpus contains 13 different datasets for ten languages in a harmonized scheme. As the CorefUD is meant to be the extension of Universal Dependencies with coreference annotation, all the datasets in CorefUD are treebanks. For some languages, human annotators provided the dependency annotations. For others, the annotation is created automatically with a parser. The coreference annotation is built upon the dependencies. This means that the mentions are subtrees in the dependency tree and can be represented with the head. In fact, in some of the datasets, there are non-treelet mentions – the mentions which do not form a single subtree. But even for these non-treelet mentions, a single headword is selected. There are some notable differences between the datasets. One of the most prominent ones is the presence of singletons. Singletons are clusters that contain only one mention. Singletons are not present in any coreference relation. However, they are annotated as mentions. For details about the dataset, please see Nedoluzhko et al. (2022) or Nedoluzhko et al. (2021). The task was simplified to predict only non-singleton mentions and group them into entity clusters.

For evaluation, the CorefUD scorer[1] is provided. The primary evaluation score is the CoNLL F_1 score with partial mention matching and singletons excluded. In the CorefUD scorer, a system mention matches a gold mention if all its words are included in the gold mention, and one of them is the key head. This means that the minimal correct span is the head, and it might be beneficial to predict mentions as only the heads.

2 Model

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

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

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

CorefUD dataset	Total size					
	docs	sents	words	empty	singletons	discont.
Catalan-AnCora	1550	16,678	488,379	6,377	74.6%	0%
Czech-PDT	3165	49,428	834,721	33,086	35.3%	3.1%
Czech-PCEDT	2312	49,208	1,155,755	45,158	1.4%	4.1%
English-GUM	150	7,408	134,474	0	75%	0%
English-ParCorFull	19	543	10,798	0	6.1%	0.7%
French-Democrat	126	13,054	284,823	0	81.8%	0%
German-ParCorFull	19	543	10,602	0	5.8%	0.3%
German-PotsdamCC	176	2,238	33,222	0	76.5%	6.3%
Hungarian-SzegedKoref	400	8,820	123,976	4,849	7.9%	0.4%
Lithuanian-LCC	100	1,714	37,014	0	11.2%	0%
Polish-PCC	1828	35,874	538,891	864	82.6%	1.0%
Russian-RuCor	181	9,035	156,636	0	2.5%	0.5%
Spanish-AnCora	1635	17,662	517,258	8,111	73.4%	0%

Table 1: Dataset Statistics

Model	Pretrained params	New params
mBERT	180M	40M
XLM-R	350M	50M

Table 2: Number of trainable parameters of the models

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

The model achieves state-of-the-art performance on the OntoNotes dataset where singletons are not annotated. We believe the model is optimal for the CorefUD dataset as well since some of the datasets of the CorefUD do not contain singletons. Moreover, the evaluation metric ignores singletons, so it does not matter that the model is not able to predict them.

Employed Models We based our model on XLM Roberta large (Conneau et al., 2020), which is significantly larger than multilingual BERT (Devlin et al., 2018) used by the baseline. The number of parameters is provided in Table 2. We also tried to use the best monolingual model for each language.

Joined Model Pretraining As you can see from Table 2, there are approximately 50 million parameters trained from scratch for XLM-R. For smaller datasets, it is practically impossible to train so many random parameters. To solve this issue, we first pre-train the model on the joined dataset and then fine-tune the model for a specific language.

Heads Prediction As mentioned above, the official scorer uses min-span evaluation with head words as min spans. Because we do not know the rules used to select single mention head in the dataset, we decided to train to model to predict the heads instead of the whole spans to optimize the evaluation metric. Having all the useful information (even dependency trees), the model should learn the original rules for selecting the head.

The simplest way to predict the mention heads would be to simply represent mention with its head word on the input. But this is not an ideal solution since multiple mentions can have the same head. If we represented a mention with only the head, some mentions would be joined, and their clusters would be merged.

To avoid this, we represent mention with the whole span, and just at the top of our model, we predict the head of each mention and output only the head word(s). This way, the mentions are represented with their spans when we build the clusters, and the clusters of two different mentions with the same head are not merged as in the case of the simple approach mentioned above.

We implemented two versions of the head prediction model. Both are implemented as separate classification heads on the top of our coreference resolution model.

The first model predicts the relative position of the head word(s) inside a span using the hidden representation of the span from the CR model. Output probabilities of head positions are obtained using sigmoid activation so the model can predict multiple heads even though there is only single head word in the gold data. This is particular optimization of the evaluation metric: If there are more words likely to be a head word of the span, it is statistically better to output all of them.

The second model uses a binary classification of each span and head candidate pair, so again, there can be more head words of a single span predicted.

Trees We believe dependency information can help the model significantly, especially when manually annotated dependencies are provided (Czech PDT, for example). Moreover, the dependency information is necessary to find mention head.

To encode syntactic information, we add to each token representation its path to *ROOT* in the dependency tree. In more detail, we first set the maximum tree depth parameter and then concatenate Bert representations of all parents up to max depth with the embedding of the corresponding dependency relation. Thus the resulting tree structure representation has the size of $max\_tree\_depth \times (bert\_emb\_size + deprel\_emb\_size)$. This representation is then concatenated with bert embedding of each token.

3 Training

We trained all the models on NVIDIA A40 graphic cards using online learning (batch size 1 document). We limit the maximum sequence length to 6 non-overlapping segments of 512 tokens. During training, if the document is longer than 6×512 tokens, a random segment offset is sampled to take random continuous block of 6 segments, and the rest of them is discarded. During prediction, longer documents are split into independent sub-documents (for simplicity, non-overlapping again). We train a model for each dataset for approximately 80k updates in our monolingual experiments. For joined-pre-trained models, we use 80k steps for model pre-training on all the datasets and approximately 30k for fine-tunning on each dataset. Each training took from 8 to 20 hours.

4 Results & Discussion

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

Monoling column shows the result of the monolingual model specific for each language. *XLM-R* column presents results of XLM Roberta large trained for each dataset separately. *Joined* is the joined model described in the Model section. The columns marked with + mean the best model from all to the left, with the additional feature. *+dev* means that the dev data part was added to training data, *+S2H* is the model with mention head prediction described earlier. Both methods for mention head prediction have statistically equal performance (we cannot tell which one is better). The reported numbers are for the first one. The results in column *+tree* correspond to adding the dependency structure as described.

It is not surprising that employing a larger model (XLMR-R large or monolingual) significantly improved the performance of the baseline. The results of the joined model are much more interesting. We can see that for some smaller datasets (e. g. German), the performance gain is huge. But if we have a look at Table 2, it makes sense because it is hard (or impossible) to train 50M parameters from scratch on a small dataset. It is also interesting that Polish is the only language where the monolingual model outperformed the joined model. But the reasons for this are probably straightforward. The polish dataset is one of the largest, so joined pre-training is not needed. Moreover there is a large monolingual model for Polish, so it is natural that it outperformed XLM-R large. For three datasets, there is a significant gain by employing mention head prediction. The difference should be even bigger when we add syntactic structure to the model. [2] Unfortunately, we did not manage to include this feature on time. From the results table, we can see that adding the trees does not help. In fact, it decreases performance significantly. We believe this is caused by some bug in our implementation, but we did not have enough time to correct it before the end of the competition.

4.1 Comparison To Other Systems

The comparison to other participating systems is shown in Table 4. Our system ended up in 3^{rd} place. Surprisingly, although the winning system outperformed ours by a large margin on average, our system reached the best performance for two datasets (*german_parcor* and *hungarian*). It would be interesting to look at both systems' differences to find out why.

[2] The potential gain by outputting only mention heads can be found in Žabokrtský et al. (2022)

Dataset/Model	monolingual model name	reference	BASELINE	Monoling	XLM-R	joined	+dev	+S2H	+Tree
ca_ancora	PlanTL-GOB-ES/roberta-base-ca	(Armengol-Estapé et al., 2021)	63.74	69.61	66.19	68.81	**70.55**	69.91	68.32
cs_pcedt	Czert-B-base-cased	(Sido et al., 2021)	70	73.74	73.55	73.85	**74.07**	71.12	73.61
cs_pdt	Czert-B-base-cased	(Sido et al., 2021)	67.27	69.81	70.99	70.63	71.49	**72.42**	70.99
de_parcorfull	deepset/gbert-base	(Chan et al., 2020)	33.75	43.04	33.75	68.91	**73.9**	68.3	65.29
de_potsdamcc	deepset/gbert-large	(Chan et al., 2020)	55.44	58.81	59.03	**70.35**	66.02	68.68	67.35
en_gum	roberta-large	(Zhuang et al., 2021)	62.59	68	66.27	68.16	**68.31**	66.88	67.39
en_parcorfull	roberta-large	(Zhuang et al., 2021)	36.44	25.84	**36.44**	30.21	31.9	23.45	40.05
es_ancora	PlanTL-GOB-ES/roberta-large-bne	(Gutiérrez-Fandiño et al., 2022)	65.98	60.12	67.99	71.24	71.48	**72.32**	72.04
fr_democrat	camembert/camembert-large	(Martin et al., 2020)	55.55	56.76	55.94	59.8	60.12	**61.39**	60.03
hu_szegedkoref	SZTAKI-HLT/hubert-base-cc	(Nemeskey, 2021)	52.35	59.76	60.68	63.24	**65.01**	64.67	62.77
lt_lcc	EMBEDDIA/litlat-bert	(Ulčar and Robnik-Šikonja, 2021)	64.81	66.93	64.81	66.34	**68.05**	67.49	64.01
pl_pcc	allegro/herbert-large-cased	(Mroczkowski et al., 2021)	65.34	**75.2**	73.19	73.66	74.46	74.56	73.38
ru_rucor	DeepPavlov/rubert-base-cased	(Kuratov and Arkhipov, 2019)	67.66	69.33	**77.5**	75.5	74.82	76.02	75.94
avg			58.53	61.30	62.03	66.21	66.94	65.94	65.94

Table 3: Results

#	User	avg	ca	cs_pcedt	cs_pdt	de_pc	de_pots	en_gum	en_pc	es	fr	hu	lt_lcc	pl_pcc	ru
1	straka	**70.72**	78.18	**78.59**	**77.69**	65.52	70.69	72.5	**39**	**81.39**	**65.27**	63.15	**69.92**	78.12	**79.34**
2	straka-single-multil	69.56	**78.49**	78.49	77.57	59.94	**71.11**	**73.2**	33.55	80.8	64.35	63.38	67.38	**78.32**	77.74
3	**ours**	67.64	70.55	74.07	72.42	**73.9**	68.68	68.31	31.9	72.32	61.39	**65.01**	68.05	75.2	77.5
4	straka-single-data	64.3	76.34	77.87	76.76	36.5	56.65	70.66	23.48	78.78	64.94	62.94	61.32	73.36	76.26
5	berulasek	59.72	64.67	70.56	67.95	38.5	57.7	63.07	36.44	66.61	56.04	55.02	65.67	65.99	68.17
6	*BASELINE*	58.53	63.74	70	67.27	33.75	55.44	62.59	36.44	65.98	55.55	52.35	64.81	65.34	67.66
7	Moravec	55.05	58.25	68.19	64.71	31.86	52.84	59.15	36.44	62.01	54.87	52	59.49	63.4	52.49
8	simple_baseline	18.14	15.58	5.51	9.48	29.81	19.41	21.99	11.37	16.64	21.74	17	27.53	15.69	24.06
9	k-sap	5.9	0	0	0	0	0	0	0	0	0	0	0	76.67	0

Table 4: Comparison to Other Participating Systems

5 Conclusion

We successfully applied an end-to-end neural coreference resolution system to the CRAC 2022 shared task. There are two main outcomes of our work. **1)** Joined training helps a lot. Our experiments support the fulfillment of the goals of the CorefUD dataset to help the models by harmonizing the annotation schemas. **2)** For the official scoring metric, predicting only the mention heads increases performance. This means that syntactic structure helps to identify mentions. Of course, such evaluation is a bit artificial and does not reflect the real-world scenario, where we do not have the gold syntax. We also suggested injecting syntactic information into the model. Unfortunately, we did not manage to get any improvement with this approach. Our system ended up in 3^{rd} place. Moreover, we reached the best performance on two datasets out of 13.

Acknowledgements

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

References

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.

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.

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

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

Asier Gutiérrez-Fandiño, Jordi Armengol-Estapé, Marc Pàmies, Joan Llop-Palao, Joaquin Silveira-Ocampo, Casimiro Pio Carrino, Carme Armentano-Oller, Carlos Rodriguez-Penagos, Aitor Gonzalez-Agirre, and

Marta Villegas. 2022. Maria: Spanish language models. *Procesamiento del Lenguaje Natural*, 68(0):39–60.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language. *arXiv preprint arXiv: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.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. Camembert: a tasty french language model. In *ACL 2020-58th Annual Meeting of the 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 Žabokrtskỳ, Amir Zeldes, and Daniel Zeman. 2022. Corefud 1.0: Coreference meets universal dependencies. In *Proceedings of LREC*.

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

Dávid Márk Nemeskey. 2021. Introducing huBERT. In *XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2021)*, page TBA, Szeged.

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.

Jakub Sido, Ondřej Pražák, Pavel Přibáň, Jan Pašek, Michal Seják, and Miloslav Konopík. 2021. Czert–czech bert-like model for language representation. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1326–1338.

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

Matej Ulčar and Marko Robnik-Šikonja. 2021. Training dataset and dictionary sizes matter in bert models: the case of baltic languages. *arXiv preprint arXiv:2112.10553*.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized bert pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227.

Zdeněk Žabokrtský, Miloslav Konopík, Anna Nedoluzhko, Michal Novák, Maciej Ogrodniczuk, Martin Popel, Ondřej Pražák, Jakub Sido, Daniel Zeman, and Yilun Zhu. 2022. Findings of the shared task on multilingual coreference resolution. In *Proceedings of the Fifth Workshop on Computational Models of Reference, Anaphora and Coreference*, Gyeongju, Republic of Korea. Association for Computational Linguistics.