Coreference Resolution for Polish: Improvements within the CRAC 2022 Shared Task

Karol Saputa
Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
karol.saputa@ipipan.waw.pl

Abstract

The paper presents our system for coreference resolution in Polish. We compare the system with previous works for the Polish language as well as with the multilingual approach in the CRAC 2022 Shared Task on Multilingual Coreference Resolution thanks to a universal, multilingual data format and evaluation tool. We discuss the accuracy, computational performance, and evaluation approach of the new System which is a faster, end-to-end solution.

1 Introduction

The paper describes our approach to coreference resolution in the Polish language submitted to the CRAC 2022 Shared Task on Multilingual Coreference Resolution.

The scope of the Shared Task was multilingual systems for 10 languages included in CorefUD 1.0 (Nedoluzhko et al., 2022). However, here we focus mainly on the improvements for the Polish language within this task and present end-to-end coreference resolution for Polish.

2 Related Work

There are two important types of references for our work: the evaluation methods for coreference resolution and previous solutions for the Polish language.

2.1 Evaluation

The popular standard for coreference resolution was created during CoNLL-2011 Shared Task as an average of MUC, B-cubed, and CEAFe scores. It is also used in the CRAC 2022 Shared Task on Multilingual Coreference Resolution.

Previous implementations included Perl script evaluation of annotation in CoNNL-U (Pradhan et al., 2014). Similarly, there is a Scoreference[1] tool Java implementation including additionally CEAFm, and BLANC, which operates on TEI (Consortium, 2022) or MMAX2 (Müller and Strube, 2006) files. It was used in the evaluation of most coreference resolution tools for the Polish language because of its compatibility with Polish Coreference Corpus (Ogrodniczuk et al., 2016) data formats.

CorefUD dataset integrates Polish Coreference Corpus and many others into one format compatible with Universal Dependencies datasets and presents a new Python reimplementation of the metric CorefUD scorer[2]. Thanks to that, there is a clear way to evaluate and compare different coreference systems.

2.2 Coreference Resolution in Polish

The current state-of-the-art solution was a Corneferencer system (Nitoń et al., 2018). It is a system based only on mention clustering i.e. it requires a text with already correctly detected mentions which are further grouped into coreference clusters and remaining singleton mentions.

The mention pairs need to have labeled heads e.g. from a dependency parsing due to input features including embedding representation of mention head token. There are other hand-crafted features e.g. mention type, mention pair distance, and mention tokens' lemmas.

It also requires the generation of mention-pairs representations which in the highest scoring version (`all2all`) results in $O(n^2)$ complexity for all mention pairs passed to the system.

The Corneferencer system achieved 81.23 F1 CoNLL (Pradhan et al., 2011) measure in the best setting during evaluation on gold mentions.

3 System description

3.1 Architecture

The submitted system is based on the `start-to-end` system (Kirstain et al.).

[1] http://zil.ipipan.waw.pl/Scoreference

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

Proceedings of the CRAC 2022 Shared Task on Multilingual Coreference Resolution, pages 18–22
October 17, 2022

This system was developed for English and is based on transformer architecture for natural language processing. It extends the Shared Task baseline system (Pražák et al., 2021) with the simplified mention-candidate representation.

3.1.1 Input features

Pre-trained model The words representation is based on the HerBERT[3] (Mroczkowski et al., 2021) pre-trained, BERT-based text encoder for the Polish language. The model has a maximum input length of 512 tokens so the longer texts are passed split (on sentence ends when possible).

End-to-end features The system works in an end-to-end fashion (Lee et al., 2017) with text-only input. In its original version (Kirstain et al.) based on the OntoNotes dataset (Weischedel et al., 2013), it included some additional annotations such as genre and speaker information which was not used here.

Such annotation is not available for the Polish dataset. Furthermore, hand-crafted features like speaker information hamper production deployment of the System.

3.1.2 Mentions

Mention representation Mention candidates are all spans of tokens (up to maximum length). Representations of candidates are based on the representation of the start and end tokens. Span representation is made to represent features related to the span is a mention.

Mention scoring Mentions are scored by calculating the biaffine combination of start and end token representations. Scores are used to prune the least scored spans from the mention candidates list.

3.1.3 Antecedents

Antecedent representation Antecedent representations are produced similarly to the mention representation except using a separate set of weights. Antecedent representation is made to represent coreference features.

Antecedent scoring Antecedents are scored by calculating the biaffine combination of two spans as concatenated start and end representations. The antecedent score measures whether two mentions are coreferent.

[3] `allegro/herbert-large-cased`

3.2 Linguistic modeling constraints

The biggest advantage of the architecture is its simplicity and low computational complexity. There are several constraints imposed by this architecture for application to Polish Coreference Corpus annotations.

3.2.1 Nested mentions

It is important for the architecture to recognize nested spans and match them with different entities. For example "the Association of Youth filmmakers" consists of two nested mentions coreferent with the association and the filmmakers. So it is needed to handle overlapping, nested spans. It is possible in `start-to-end` architecture by including all possible spans.

3.2.2 Singleton and mention head recognition

Polish Coreference Corpus includes annotation of singletons - mentions that have no coreferent mentions.

Scoring during the CRAC 2022 Shared Task on Multilingual Coreference Resolution omits singletons. `Start-to-end` architecture does not detect singletons as the spans are scored for the antecedent relation in pairs and it is the only element of the loss function (and model optimization). Singletons may not be included in the detected mentions since they should not be considered in antecedent scoring.

Including singletons in the task would need a modification of the loss function or adding an additional model.

3.3 Data augmentation

Polish Coreference Corpus consists of about 1800 documents consisting of one or more paragraphs of text, each originating from one source. Samples used for training included the original texts and subsamples.

Paragraphs and pairs of sentences were treated as additional separate subsamples that can be added to training samples. The coreference annotation was filtered to include only relations inside the sample.

The process of augmentation was controlled by parameters of a fraction of sentence pairs and paragraphs to include in the training sample.

Using samples of shorter lengths was important to improve performance on short texts.

3.4 Training

Dynamic batching There was dynamic batching

System	Precision	Recall	F1
submission	88.11	71.22	78.77
herbert–base	86.83	75.33	80.67
herbert–large	86.26	80.60	83.33

Table 1: Mention detection F1 measure results for Polish on the development set, singletons excluded.

System	F1
submission	63.64
herbert–base	72.44
herbert–large	73.39
corneferencer	82.44

Table 2: CoNLL F1 measure results for coreference resolution in Polish on the development set, singletons excluded.

applied - a constant maximum total batch length of texts. It was important in batching samples of different sizes e.g. short and long texts, and sentence pairs.

Optimization Model was optimized using PyTorch `AdamW` implementation with learning rate (1e-5), linear decay, and warm-up steps (5000) as recommended in `start-to-end` implementation[4].

4 Results

We compare metrics speed for the System with the Corneferencer and other submissions.

4.1 Performance

4.1.1 Mention detection

Mention detection is an important element of the system. Lack of detected spans impacts coreference resolution measures. Results are presented in Table 1.

Redundant spans do not lower performance because they can be assigned no coreference relation (null span antecedent). It corresponds to the higher precision of the system. Improving mention detection could be the first element of the overall improvement.

4.1.2 Coreference Resolution

Corneferencer comparison The previous solution for Polish, Corneferencer, was tested on gold mention annotation because the mentions are needed to process texts with this tool and used available model[5] , thus a different subset of PCC was used for comparison in Table 2, 200 texts from the test split used in Corneferencer evaluation.

Pre-trained models We compared the base (12 layers, `herbert-base`) and large (24 layers, `herbert-large`) version of the pre-trained encoder used in the System. The results are presented in Table 2. The smaller model was trained 71 000 steps and the larger one with 45 000 steps. The larger model gave a 1.31% improvement, with a 1.7% increase gained in the last 10 000 steps (F1 difference between 35 000 training steps and the final one). One step is one optimization step of the model.

Training step	Train F1	Dev F1	Difference
1000	1.56	0.87	0.69
5000	26.46	24.72	1.74
10000	58.73	55.45	3.28
15000	77.65	66.10	11.55
20000	84.81	69.31	15.50
25000	89.96	71.40	18.56
30000	92.90	72.10	20.81
35000	95.01	72.24	22.77
40000	96.03	72.63	23.40
45000	96.88	73.46	23.43

Table 3: Comparison of the development set generalization of the System during training, F1 evaluation of training and development sets.

4.2 Development set generalization

Comparison of the development set generalization of the System during training is presented in Table 3. As presented in (Yang et al.), it is a behavior of the big models, such as BERT-based models, to overcome the bias-variance tradeoff. The increasing difference between training and development sets does not impact model generalization.

4.3 Multilingual generalization

The System was tested on other languages in the Shared Task to test the degree of performance drop in such a zero-shot setting. Results are presented in Table 4. There was no attempt to use a multilingual pre-trained model or training on the other languages. The best result, 41.84, was achieved on the English dataset, the architecture used in the System was initially used for this language.

[4] `https://github.com/yuvalkirstain/s2e-coref`

[5] `http://zil.ipipan.waw.pl/Corneferencer?action=AttachFile&do=view&target=model_1190_features.h5`

Dataset	F1
en_parcorfull	22.34
de_parcorfull	13.67
lt_lcc	21.91
en_gum	41.84
es_ancora	21.87
fr_democrat	0.0
cs_pcedt	23.67
cs_pdt	27.94
ru_rucor	17.88
ca_ancora	17.49
pl_pcc	76.67
de_potsdamcc	40.59
hu_szegedkoref	11.45
average	25.95

Table 4: CoNLL F1 measure results for the System for all languages - trained only on Polish corpus with pre-trained model for Polish. Value for fr_democrat was not calculated due to technical issues.

System	Time [s]
herbert–large (GPU)	0.0542
herbert–large (CPU)	0.1845
corneferencer	271.7

Table 5: Document processing time - comparison of processing speed between `start-to-end` architecture and Corneferencer - previous solution for Polish. The

4.4 Processing speed

For the comparison of the System with the previous solution for Polish an important aspect is also the processing speed. Table 5 presents the results of comparison for Corneferencer and GPU/CPU versions of the System. Corneferencer time was calculated as a mean of two executions for three randomly chosen texts, and the System time was calculated as a mean over the test set.

Time included in the Corneferencer processing does not include e.g. mention detection. It is not a total time needed for the coreference resolution task and still, it is three orders of magnitude longer.

4.5 Submission

The model submitted to the Shared Task achieved a score of 76.67 F1 measure on the Polish test set. The submission was named `k-sap`. It was not the best result for Polish in the competition. It was overtaken by `straka` (78.12 F1, 1.019% improvement) and `straka-single-multilingual-model` (78.32 F1, 1.022%) which were multilingual submissions.

Submission	F1 Polish
straka-single-multilingual-model	78.32
straka	78.12
k-sap	76.67
ondfa	75.20
straka-only-single-treebank-data	73.36

Table 6: CRAC 2022 Shared Task on Multilingual Coreference Resolution Evaluation results for Polish, top 5 results.

The submitted model was undertrained (Section 4.2), and the train-dev difference was 9.77 F1 points. The results of the submission model on the Corneferencer dataset are lower (Table 2). There could have been test data leakage from original TEI files which we did not think was possible during the submission phase.

5 Further Work

Longformer There is a Longformer model for Polish available on Hugging Face Models[6]. It could improve results for longer texts (which are included in the Polish test set). However, it is not popular yet and was not tested.

Multilingual comparison 4 Shared Task submissions achieved above 60 F1 score, all of which gained more than 70 F1 for the Polish test subset. A comparison of these methods should help to answer questions: (1) is there still a need for a language-specific solution, (2) whether there are issues with data quality between corpora for different languages that could be improved by using guidelines from top-scored datasets.

6 Summary

For Polish, the System is faster, end-to-end, and has comparable performance to the previous solution.

There is a need to analyze other submissions to assess the state of language-specific systems' performance, however, we see that there is a capability to build a high-performing multilingual system.

The presence of a multilingual dataset and evaluation tool provides the infrastructure to build such a system efficiently and track progress.

[6] `sdadas/polish-longformer-large-4096`

Acknowledgements

This work was supported by the European Regional Development Fund as a part of 2014–2020 Smart Growth Operational Programme, CLARIN — Common Language Resources and Technology Infrastructure, project no. POIR.04.02.00-00C002/19 and by the project co-financed by the Minister of Education and Science under the agreement 2022/WK/09.

References

TEI Consortium. 2022. TEI P5: Guidelines for Electronic Text Encoding and Interchange. Publisher: Zenodo Version Number: v4.4.0.

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

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

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.

Christoph Müller and Michael Strube. 2006. Multi-level annotation of linguistic data with MMAX2. In Sabine Braun, Kurt Kohn, and Joybrato Mukherjee, editors, *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, pages 197–214. Peter Lang, Frankfurt a.M., Germany.

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.

Bartłomiej Nitoń, Paweł Morawiecki, and Maciej Ogrodniczuk. 2018. Deep neural networks for coreference resolution for Polish. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Maciej Ogrodniczuk, Katarzyna Głowińska, Mateusz Kopeć, Agata Savary, and Magdalena Zawisławska. 2016. Polish Coreference Corpus. In *Human Language Technology. Challenges for Computer Science and Linguistics*, Lecture Notes in Computer Science, pages 215–226, Cham. Springer International Publishing.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–35, Baltimore, Maryland. Association for Computational Linguistics.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. CoNLL-2011 Shared Task: Modeling Unrestricted Coreference in OntoNotes. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–27, Portland, Oregon, USA. Association for Computational Linguistics.

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

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes Release 5.0. Artwork Size: 2806280 KB Pages: 2806280 KB Type: dataset.

Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking bias-variance trade-off for generalization of neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, pages 10767–10777. PMLR. ISSN: 2640-3498.