Deep Neural Networks for Coreference Resolution for Polish

Bartłomiej Nitoń, Paweł Morawiecki, and Maciej Ogrodniczuk
Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
{bartek.niton, pawel.morawiecki}@gmail.com, maciej.ogrodniczuk@ipipan.waw.pl

Abstract

The paper presents several configurations of deep neural networks aimed at the task of coreference resolution for Polish. Starting with the basic feature set and standard word embedding vector size we examine the setting with larger vectors, more extensive sets of mention features, increased number of negative examples, Siamese network architecture and a global mention clustering algorithm. The highest results are achieved by the system combining our best deep neural architecture with the sieve-based approach – the cascade of rule-based coreference resolvers ordered from most to least precise. All systems are evaluated on the data of the Polish Coreference Corpus featuring 540K tokens and 180K mentions. The best variant improves the state of the art for Polish by 0.53 F1 points, reaching 81.23 points of the CoNLL metric.

Keywords: coreference resolution, deep neural network, Polish

1. Introduction

Coreference resolution, the task of clustering textual fragments that refer to the same entity in the discourse world, has been successfully tackled for Polish in numerous configurations, starting with a rule-based model (Ogrodniczuk and Kopeć, 2011) through machine-learning (Kopeć and Ogrodniczuk, 2012) and projection-based approaches (Ogrodniczuk, 2013) up to the newest multi-pass sieve setting (Nitoń and Ogrodniczuk, 2017). In this paper we present the first deep neural network resolver for Polish, further improving the state of the art. For English, the state-of-the-art coreference resolution systems are also based on deep neural networks (Clark and Manning, 2016), (Wiseman et al., 2016). We were inspired and motivated by these works.

The data for our experiments, as for all previous configurations, come from the Polish Coreference Corpus (Ogrodniczuk, 2015, PCC), a large corpus of Polish general nominal coreference manually annotated over the texts of the National Corpus of Polish (Przepiórkowski et al., 2012)[1] and Rzeczpospolita Corpus (Presspublica, 2002). The corpus features broad understanding of mentions (e.g. with included relative clauses or appositions, nesting, discontinuities and zero anaphora) and contains almost 1800 documents from 14 genres, 540K tokens, 180K mentions and 128K coreference clusters.

Coreference scores on the test set are measured using gold mentions on input with *MUC* (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), and *CEAFE* (Luo, 2005) metrics averaging them according to the CoNLL-2011 approach (Pradhan et al., 2011) to track influence on different coreference dimensions (the B^3 measure being based on mentions, *MUC* on links, and *CEAFE* on entities). *CEAFM* (Luo, 2005) and *BLANC* (Recasens and Hovy, 2011) are also presented for consideration. Metrics were calculated using *Scoreference*[2], a mention detection and coreference resolution evaluation tool (Ogrodniczuk et al., 2015).

[1] Pol. Narodowy Korpus Języka Polskiego (NKJP), see `http://nkjp.pl`.

[2] `http://zil.ipipan.waw.pl/Scoreference`

2. The Baseline

In all our experiments we used 90% texts from the *PCC* as the training set and 10% as the test set. Text type balance was maintained in this division.

Our neural networks return a single output (a value between 0 and 1), which is interpreted as the probability of two mentions being coreferent. Mentions are then linked into coreference chains with a certain *clustering algorithms*. We experimented with both *mention-based* and *entity-based* settings. The *mention-based* algorithm connects each anaphor with an antecedent for which neural network returned the best prediction score. The *entity-based* algorithm connects each anaphor with a mention group for which the neural network returned the best average prediction score (which is average prediction between the anaphor and each mention being part of the tested mention group).

For both types of algorithms the prediction must be higher than selected *connection threshold*, i.e. the value above which two mentions are considered coreferent. Each experiment (excluding *Experiment 3*) was tested on a set of various different pre-selected *connection threshold* values: *0.5*, *0.75*, *0.85*, *0.95*, and *0.99*.

2.1. Input Features

Each training features vector gathers information about antecedent, anaphor and antecedent-anaphor pair. Each mention features vector consists of:

- word embedding vectors (Wawer, 2015) for the mention head word, the first word in the mention, two words preceding the mention and two words following the mention
- averages of embeddings vectors calculated for five words preceding the mention, five words following the mention, the words of the mention and the words of the sentence in which the mention occurred
- binary features marking whether the mention is of a

nominal type[3], pronominal type[4], a zero type[5] or other.

Each pair's features vector consists of the distances between mentions in the pair measured in words and in mentions[6], and a set of binary features marking whether:

- mentions in a pair intersect
- mentions are identical (two features: without lemmatization or using lemmatized mentions strings, obtained with Morfeusz morphological analyser[7] (Woliński, 2014) and Pantera tagger[8] (Acedański, 2010))
- mentions are in the same sentence
- mentions are in the same paragraph
- one mention is an acronym of the other
- the antecedent contains the rarest (in terms of frequency) word from the anaphora[9].

In this experiment we used the word embedding vectors of the size 50. Each training example (pair of mentions) has 1147 features (554 for each mention and 39 pair features describing their relations) and is labeled with 1 or 0 marking whether mentions are coreferent or not.

2.2. Network Architecture

Input features described above are concatenated into a single vector and act as input to our neural network. Thus, the network takes an input vector of 1147 units and is passed through a fully connected network with a single output (a value between 0 and 1). The output is interpreted as the probability of two mentions being coreferent. The network has 3 hidden layers, where a number of units in subsequent layers are 500, 300, and 100. In hidden layers we use RECTIFIED LINEAR UNIT — RELU (Nair and Hinton, 2010) as an activation function and a sigmoid function in the output layer.

2.3. Training Details

The network is trained by finding the parameters (weights) to minimize the loss function. Regarding the loss, we follow a typical choice, namely a binary cross entropy function. During the training, the loss was minimized with ADAM (Kingma and Ba, 2014) for 2 epochs with mini-batches of size 128. We experimented with longer training (more epochs) but the network became overfitted. We used batch normalization (Ioffe and Szegedy, 2015) in each hidden layer and the network was regularized using dropout (Srivastava et al., 2014) with a rate of 0.2.
Part of input features consists of word embeddings and these vectors are treated as static and are not modified during training.
Training set had 426 thousands pairs of mentions, equally split between positive and negative pairs. The neural network model was implemented with KERAS (Chollet and others, 2015) using TENSORFLOW (Abadi et al., 2016) as a backend. For training we used the GPU (K40 TESLA) and the training was completed within a few minutes (around 2 minutes per epoch). The implemented models are publicly available at `http://zil.ipipan.waw.pl/Corneferencer`.

2.4. The Results

First we evaluated the neural network model on the test set consisting of 40K mention pairs. Our baseline model accuracy is 72.27%, which means approximately 72% of examples are classified correctly.
Then we evaluated the neural network on whole texts (not only selected mention pairs) from the test set using THE CORNEFERENCER[10] system specially implemented for this task. The best score was acquired for *the mention-based* clustering algorithm with *the connection threshold* 0.99 (see row labeled as *Baseline* in Table 1 for results).

3. The Experiments

3.1. Experiment 1: Larger Vectors

After experimenting with the basic feature set we tested different architectures in pursuit of a better, more robust model. The first improvement featured larger word embedding vectors (of the size 300 instead of 50), which gave 6647 features for each training example. However, despite much richer embeddings, we did not observe any significant improvements in the evaluation metrics. The best results were acquired for *mention-based* clustering algorithm with 0.99 *connection threshold* (see *Experiment 1* in Table 1). It might be the case that 50-value embeddings are just enough to capture similarities (or any other relations) relevant to our task.

3.2. Experiment 2: More Features

In the next step we brought back embeddings vector size to 50 and added extra input features to the training examples. We selected the features proved best in other coreference resolution systems for Polish, e.g. the model described in

[3]Nominal mentions are all nominal phrases whose syntactic head is a noun marked with a `subst` (general noun) or `ger` (gerund) tags (see `http://nkjp.pl/poliqarp/help/en.html` for a concise tag descriptions).

[4]Pronominal mentions are first-, second- (annotated as `ppron12`) or third-person pronouns (`ppron3`).

[5]Zero mentions are marked with tags corresponding to verbal forms (`fin`, `praet`, `bedzie`, `winien`, `aglt`, and `impt`).

[6]Distances are binned into one of the buckets [0,1,2,3,4,5-7,8-15,16-31,32-63,64+,discontinuous] and then represented as binary features (last bucket is reserved for situation when one mention is between parts of second discontinuous mention)

[7]`http://sgjp.pl/morfeusz/`

[8]`http://zil.ipipan.waw.pl/PANTERA`

[9]For the purpose of checking word rarity we used a word frequency list extracted from the balanced subcorpus of the National Corpus of Polish (Przepiórkowski et al., 2012).

[10]CORNEFERENCER (`http://zil.ipipan.waw.pl/Corneferencer`) is a neural network based tool for performing coreference resolution. It is the final product of the research described in this article, it was used to get system annotation for each experiment using for this task pretrained neural networks. Default CORNEFERENCER configuration is the one described as *Experiment 5*.

(Ogrodniczuk et al., 2015) and in (Nitoń and Ogrodniczuk, 2017).
Additional binary mention features are e.g. features marking whether the mention:

- is in first or second person
- starts with a demonstrative pronoun
- starts with a demonstrative pronoun and is nominal
- starts with a demonstrative pronoun and is pronominal or zero
- is a reflexive pronoun
- is first in a sentence
- is a personal pronoun or zero mention (false, if not one of them)
- head contains a digit
- contains a letter
- is post modified (a head word is not the last word in the mention).

Additional binary pair features are features marking whether:

- distance between mentions in sentences is 1, 2 or more (3 features)
- their gender values agree (without distinction of masculine gender into subtypes)
- the string of one mention starts with second mention's string
- the string of one mention ends with second mention's string
- the string composed of the initial letters of all the capitalized words in the mention string produces a string matching a head word of the second mention
- mentions are in the same sentence, the anaphor is pronominal, and the antecedent is the first in paragraph
- mentions are in the same sentence, their persons and numbers agree, and the antecedent is the first in paragraph
- mentions are in adjacent sentences, are adjacent mentions (without any other mention in between), their persons and numbers agree and the anaphor is pronominal
- mentions are in adjacent sentences, are adjacent mentions and the anaphor is pronominal
- they safisfy additional conditions for six knowledge-based features — 3 PLWORDNET-based and 3 WIKIPEDIA-based, closely described in (Ogrodniczuk et al., 2015).

We also added string kernel features matching whole mentions or their heads (2 features).
As suspected, the features which are working well in other systems also significantly increased the evaluation metrics of our solution. Best results were acquired for *mention-based* clustering algorithm with 0.95 *connection threshold* (see *Experiment 2* in Table 1).

3.3. Experiment 3: Siamese Networks

Next we tried a different network architecture called the Siamese network (Bromley et al., 1994). Networks of this type are particularly useful for tasks that involve finding similarity or a relationship between two comparable things. The network consists of two identical subnetworks (weights are shared) to process two inputs followed by another module which produces the final output. We used here same embeddings vector size and features as in *Experiment 2* with the difference that one network uses all mention features of the antecedent and features corresponding to the tested mention pair and the other uses all mention features for the anaphora and also mention pair features. So we are using same pair features at the input of both networks.
Typically, Siamese networks are applied to determine whether two faces belong to the same person or to figure out whether two signatures come from the same person. Unfortunately, this architecture did not bring us any improvement over the baseline results (see *Experiment 3* in Table 1 for the best acquired, in this experiment, results).

3.4. Experiment 4: More Negative Examples

In the next experiment we used features, embeddings vector size, and architecture from *Experiment 2* but extended the training set by additional 600 thousands negative pairs of mentions, also including singletons. Dominance of negative examples over positive is a typical situation in real texts, where most pairs are not coreferent. Thus our new training set should correspond better to a real test scenario. The best results were obtained for *mention-based* clustering algorithm with 0.85 *connection threshold* (see *Experiment 4* in Table 1) and improve the metrics by over 3%.

3.5. Experiment 5: *All2all* Mention-based Clustering Algorithm

The *mention-based* detection algorithm, in its base form, considers only mentions preceding the mention to be clustered. In this experiment we checked all possible mention pairs regardless their positions in the text. We used here the same configuration (embeddings vector size, network architecture, features) as in *Experiment 4*.
Best results were acquired for 0.85 *connection threshold* (see *Experiment 5* in Table 1). We refer later to this clustering algorithm as *all2all*.

3.6. Experiment 6: Mixed Architecture

In the last experiment we simulated mixing the sieve-based architecture described in (Nitoń and Ogrodniczuk, 2017) with our best neural system configuration (*Experiment 5*). To acquire this we preprocessed input data with the sieve-based coreference resolver using different sieve configurations and then by CORNEFERENCER tool using the *all2all*

System	MUC [%]			B3 [%]			CEAFM [%]		
	P	R	F1	P	R	F1	P	R	F1
Baseline	71.87	40.15	51.52	**94.87**	79.35	86.42	78.65	78.65	78.65
Experiment 1	64.96	44.46	52.79	91.72	80.43	85.71	77.79	77.79	77.79
Experiment 2	62.80	59.30	61.00	87.64	83.72	85.64	78.85	78.85	78.85
Experiment 3	55.67	55.07	55.37	84.41	82.39	83.39	75.69	75.69	75.69
Experiment 4	**72.64**	59.66	65.51	91.08	83.95	**87.37**	**82.08**	**82.08**	**82.08**
Experiment 5	69.31	65.28	67.23	87.19	86.01	86.59	81.14	81.14	81.14
Experiment 6	70.34	**68.12**	**69.21**	86.76	**86.72**	86.74	81.69	81.69	81.69

System	CEAFE [%]			BLANC [%]			CoNLL
	P	R	F1	P	R	F1	[%]
Baseline	77.02	**90.37**	83.16	**85.08**	60.08	65.42	73.70
Experiment 1	77.99	87.66	82.54	78.23	62.91	67.59	73.68
Experiment 2	82.76	84.57	83.65	76.57	68.16	71.54	76.76
Experiment 3	81.19	81.54	81.37	70.54	66.13	68.05	73.37
Experiment 4	84.33	90.24	87.19	76.97	69.65	**72.70**	80.02
Experiment 5	85.92	87.88	86.89	68.45	74.03	70.85	80.24
Experiment 6	**87.21**	88.29	**87.75**	68.10	**74.71**	70.86	**81.23**

Table 1: Comparison of coreference resolution scores for different experiments with neural networks

System	MUC [%]			B3 [%]			CEAFM [%]		
	P	R	F1	P	R	F1	P	R	F1
Ruler	51.38	65.61	57.63	78.78	84.99	81.76	74.57	74.57	74.57
Bartek–3	61.14	67.90	64.34	84.08	86.09	85.07	79.81	79.81	79.81
Bartek–S1	70.30	65.35	67.73	**87.91**	85.38	86.63	**81.74**	**81.74**	**81.74**
Neural	69.31	65.28	67.23	87.19	86.01	86.59	81.14	81.14	81.14
Mixed	**70.34**	**68.12**	**69.21**	86.76	**86.72**	**86.74**	81.69	81.69	81.69

System	CEAFE [%]			BLANC [%]			CoNLL
	P	R	F1	P	R	F1	[%]
Ruler	84.89	75.65	80.00	70.69	68.53	69.55	73.13
Bartek–3	86.99	83.22	85.06	**75.67**	73.01	**74.26**	78.16
Bartek–S1	86.56	**88.96**	87.74	70.19	71.73	70.93	80.70
Neural	85.92	87.88	86.89	68.45	74.03	70.85	80.24
Mixed	**87.21**	88.29	**87.75**	68.10	**74.71**	70.86	**81.23**

Table 2: Comparison of coreference resolution systems

clustering algorithm. As we can see in Table 1 it brings some improvement for coreference resolution even over sieve-based solution (see Table 2). We think that is due to the fact that such system uses more complex mechanisms in cases where simple rules fail. It also merges initial (detected by sieve system) mention groups by hardest links between their mentions based on the prediction made by the neural network.

Best results were acquired while preprocessing data with full set of sieves described in (Nitoń and Ogrodniczuk, 2017) as best configuration and 0.95 *connection threshold* (see *Experiment 6* in Table 1).

4. Summary

Table 2 presents comparison of our new coreference resolution strategies (*Neural* and *Mixed*) with *Bartek–S1*, sieve-based solution described in (Nitoń and Ogrodniczuk, 2017) and two existing coreference resolution systems for Polish described in detail in (Ogrodniczuk et al., 2015). RULER is a simple rule-based tool with design following (Haghighi and Klein, 2007) and BARTEK–3 is an adaptation of the BART system for Polish, being a machine learning-based solution.

The comparison shows that using solely neural network-based system we can almost reach the state of the art for coreference resolution score for Polish. Combining the sieve-based architecture and the best acquired neural network configuration has led to the best score for Polish coreference resolution ($\tilde{0}.5\%$ improvement in *CoNLL* over the best sieve-based system). We think that there is still room for improvement, specifically by trying different neural architectures and/or using knowledge from sieves in the training phase of a neural net. The main disadvantage of using neural networks is the clustering time, which is way longer than in compared approaches, therefore it is not the best solution for real-time working tools.

5. Acknowledgements

The work reported here was carried out within the research project financed by the Polish National Science Centre (contract number 2014/15/B/HS2/03435).

6. Bibliographical References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283.

Acedański, S. (2010). A Morphosyntactic Brill Tagger for Inflectional Languages. In Hrafn Loftsson, et al., editors, *Advances in Natural Language Processing: Proceedings of the 7th International Conference on NLP (IceTAL 2010)*, pages 3–14. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bagga, A. and Baldwin, B. (1998). Algorithms for Scoring Coreference Chains. In *Proceedings of The 1st International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566.

Bromley, J., Guyon, I., Lecun, Y., Säckinger, E., and Shah, R. (1994). Signature Verification using a "Siamese" Time Delay Neural Network. In *In NIPS Proc*.

Chollet, F. et al. (2015). Keras. `https://github.com/fchollet/keras`.

Clark, K. and Manning, C. D. (2016). Deep reinforcement learning for mention-ranking coreference models. *arXiv preprint arXiv:1609.08667*.

Haghighi, A. and Klein, D. (2007). Unsupervised Coreference Resolution in a Nonparametric Bayesian Model. In John A. Carroll, et al., editors, *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 848–855. The Association for Computational Linguistics.

Ioffe, S. and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Francis R. Bach et al., editors, *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org.

Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.

Kopeć, M. and Ogrodniczuk, M. (2012). Creating a Coreference Resolution System for Polish. In Nicoletta Calzolari, et al., editors, *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 192–195, Stambuł. European Language Resources Association.

Luo, X. (2005). On Coreference Resolution Performance Metrics. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 25–32, Vancouver, Canada. Association for Computational Linguistics.

Nair, V. and Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. In Johannes Fürnkranz et al., editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814. Omnipress.

Nitoń, B. and Ogrodniczuk, M. (2017). Multi-pass sieve coreference resolution system for Polish. In Jorge Gracia, et al., editors, *Proceedings of the 1st Conference on Language, Data and Knowledge (LDK 2017)*, number 10318 in Lecture Notes in Artificial Intelligence, pages 222–236. Springer International Publishing, Berlin.

Ogrodniczuk, M. and Kopeć, M. (2011). Rule-based coreference resolution module for Polish. In *Proceedings of the 8th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC 2011)*, pages 191–200, Faro, Portugal.

Ogrodniczuk, M., Głowińska, K., Kopeć, M., Savary, A., and Zawisławska, M. (2015). *Coreference in Polish: Annotation, Resolution and Evaluation*. Walter De Gruyter.

Ogrodniczuk, M. (2013). Translation- and Projection-Based Unsupervised Coreference Resolution for Polish. In Mieczysław A. Kłopotek, et al., editors, *Proceedings of the 20th International Conference Intelligent Informa-*

tion Systems, volume 7912 of *Lecture Notes in Computer Science*, pages 125–130. Springer-Verlag, Berlin, Heidelberg.

Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R., and Xue, N. (2011). CoNLL-2011 Shared Task: Modeling Unrestricted Coreference in OntoNotes. In *Proceedings of the 15th Conference on Computational Natural Language Learning: Shared Task*, CONLL Shared Task '11, pages 1–27, Stroudsburg, PA, USA. Association for Computational Linguistics.

Adam Przepiórkowski, et al., editors. (2012). *Narodowy Korpus Języka Polskiego [Eng.: National Corpus of Polish]*. Wydawnictwo Naukowe PWN, Warsaw.

Recasens, M. and Hovy, E. H. (2011). BLANC: Implementing the Rand Index for Coreference Evaluation. *Natural Language Engineering*, 17:485–510.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Vilain, M., Burger, J., Aberdeen, J., Connolly, D., and Hirschman, L. (1995). A Model-Theoretic Coreference Scoring Scheme. In *Proceedings of the 6th Message Understanding Conference (MUC-6)*, pages 45–52.

Wawer, A., (2015). *Sentiment Dictionary Refinement Using Word Embeddings*, pages 186–193. Springer International Publishing, Cham.

Wiseman, S., Rush, A. M., and Shieber, S. M. (2016). Learning Global Features for Coreference Resolution. *arXiv preprint arXiv:1604.03035*.

Woliński, M. (2014). Morfeusz reloaded. In Nicoletta Calzolari, et al., editors, *Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014*, pages 1106–1111, Reykjavík, Iceland. ELRA.

7. Language Resource References

Maciej Ogrodniczuk. (2015). *Polish Coreference Corpus*. Institute of Computer Science, Polish Academy of Sciences, `http://zil.ipipan.waw.pl/PolishCoreferenceCorpus`.

Presspublica. (2002). *Rzeczpospolita Corpus*. Dawid Weiss, `http://www.cs.put.poznan.pl/dweiss/rzeczpospolita`.