Retraining and Evaluation of an off-the-shelf Coreference Resolution system

Iana Palacheva
University of Potsdam
iana.palacheva@uni-potsdam.de

Polina Gusenkova
University of Potsdam
polina.gusenkova@uni-potsdam.de

Karina Hensel
University of Potsdam
karina.hensel@uni-potsdam.de

Abstract

A crucial subtask many modern Natural Language Processing (NLP) applications is coreference resolution, which is the correct mapping of expressions in a text to the actual entities, which they refer to. With the recent advances in machine-learning various neural systems have been developed for this purpose. Many of them are freely available and require only few computational resources, which makes them easy to integrate into existing NLP pipelines. However, most of these models are trained and evaluated on English language data only. Therefore, in this project we would like to analyse the performance of one such pre-defined, open-source system for Russian data. Moreover, our goal is to evaluate the original English system.

1 Introduction

Coreference resolution is the task of linking all so called mentions in a text to the real-world entities, which they refer to. Mentions can be of three types: pronouns (e.g. 'he'), named entities (e.g. 'Big Ben') or noun phrases (NPs) (e.g. 'the black cat'). This is slightly different from the (purely linguistic) concept of anaphora resolution, which connects a mention to all previous mentions in a text, which refer to it (Manning, 2019).

Natural language processing (NLP) is mainly concerned with coreference resolution although both terms are often used interchangeably. Resolving relations between mentions and entities correctly is an integral part of many NLP tasks. For instance, a successful machine translation application also requires a precise coreference resolution system in its pipeline. Other examples are chat-bots, question answering systems, and other applications that involve information extraction.

The challenge is to find a trade-off between accuracy and usability. There exist already pretrained coreference systems, which are easy to integrate into ones' own applications. An example is the 'NeuralCoref'[1] model (Wolf et al., 2020), which has been trained on the English 'OntoNotes 5.0' corpus (Weischedel et al., 2013). Like most other coreference systems nowadays, it builds on neural networks.

One main advantage of learning-based models over rule-based coreference resolution systems is that mention features are inferred implicitly during training. Hence, no manual, language-specific feature engineering is necessary. Therefore, in theory, neural systems can be trained and evaluated on multiple languages without much effort. The system architecture remains unchanged, only the input vectors differ.

Therefore, neural systems can be trained and evaluated on multiple languages without much effort. The system architecture remains unchanged, only the input vectors differ.

Neural coreference resolution systems can further be divided into entity-mention / mention-pair (e.g. Durrett and Klein (2014)), clustering-based (e.g. Clark and Manning (2016b)) and mention-ranking models (e.g. Wiseman et al. (2015)). Mention-ranking systems compute a score for a pair of mentions to be coreferent. They are less accurate than entity-based systems because they lack the notion of entities and world-knowledge. They model coreference resolution as a sequence of independent decisions instead of comparing and ranking possible coreference clusters or chains. Nevertheless, mention-ranking models are usually faster and easier to train, which makes them quite useful when one wants to integrate a coreference resolution model in an existing application

[1] https://github.com/huggingface/neuralcoref

pipeline.

However, most of these models have only been developed for and evaluated on English data. Therefore, one research question which we would like to raise is how a neural model, which has primarily been developed for and trained on English, performs on other language data.

Other aspect we are interested in are usability, extensibility, and whether off-the-shelf, rather light-weight and generic systems perform equally well for linguistically different languages. To this end, we attempt to retrain the NeuralCoref model for Russian. Russian data annotated with coreference is taken from the RuCor[2] (Toldova et al., 2014) corpus. Additionally, we evaluate the original English model as a separate task initially intended for comparison with the retrained Russian model. We use the MUC, $CEAF_e$ and B^3 metrics for evaluation.

As a result of our efforts, we demonstrate significant issues with usability, extensibility, and reproducibility of the existing off-the-shelf systems on the example of NeuralCoref. With this report, we would like to highlight that seemingly simple and well-described instructions do not solve the problem of under-documented and not unified enough data. We hope that this report would inspire developers of open-source software to work towards better flexibility in terms of possible changes, especially when these changes are assumed and encouraged.

2 Previous and related work

There exist two main types of approaches to coreference resolution in NLP: rule-based systems (e.g. Hobbs (1978), which use a set of handcrafted, often language-specific rules to identify coreference relations, and learning-based system, which infer these features rather implicitly during training. Examples of machine-learning coreference resolution systems have been developed by Durrett and Klein (2013) and Wiseman et al. (2015). Learning-based models are usually faster and often times more accurate than the traditional rule-based systems but they also require a sufficient amount of data for training and testing. This is especially a problem for under-resourced languages and also used to be an issue for Russian.

With the release of the *RuCor* corpus (Toldova et al., 2014), however, there is now available a data set annotated for coreference relations between NPs which can be used for the implementation and evaluation of machine-learning coreference systems in Russian. A detailed description of the corpus can be found in section 4.1.

One such learning-based model has been created by Kupriianova et al. (2018). It largely adapts the design of the mention-ranking model proposed by Clark and Manning (2016a). Although mention-ranking systems, which rank possible mention-antecedent pairs rather than comparing entire coreference chains, generally perform worse than cluster-ranking models, Kupriianova et al. (2018) reach a F1-score of 0.7131 on the *RuCor* corpus, which is about 8 percent higher than the previous state-of-the-art reported in Toldova et al. (2014). However, even though though the source code for this research is published, the system is not designed to be usable as a sub-module of an NLP pipeline.

There are several major multilingual systems out there, such as CorefAnnotator for English and Chinese from Stanford CoreNLP[3], Coreference Resolution module from AllenAI[4] and xrenner[5] - externally configurable coreference and non named entity recognizer which employs both machine learning and declarative programming principles and currently configured for English, German, Hebrew and Coptic languages.

The system which we will use in this project adapts the approach proposed by Clark and Manning (2016a) too: NeuralCoref. Since we are focused on usable systems, it meets our research goals: NeuralCoref is an open-source actively developed system which is production-ready and optimized for simple usage. Moreover, it is accompanied by explicit instructions on training which makes it the most suitable candidate for experimental adaptation for other language. The following section describes the system in further detail.

[2] http://rucoref.maimbava.net/

[3] https://stanfordnlp.github.io/CoreNLP/coref.html

[4] https://demo.allennlp.org/coreference-resolution/coreference-resolution

[5] https://corpling.uis.georgetown.edu/xrenner

3 NeuralCoref system

3.1 General architecture

NeuralCoref extends the SpaCy2.1+ pipeline to identify and annotate co-reference clusters by using neural networks. It is a mention-ranking model already pre-trained on the OntoNotes 5.0 (Weischedel et al., 2013) corpus which contains web pages and news articles. Therefore, at evaluation time one has to keep in mind that the system incorporates some bias towards more formal language.

The model consists of two parts: an initial embedding layer concatenates the averaged word vectors of the tokens inside a mention (when it is a multi-word expression, e.g. *Hillary Clinton* is one mention consisting of two tokens) and those of the surrounding textual content with additional features (mention length, speaker position info etc.) to obtain a full representation of a mention and its context (Figure 1)natbibnatbib.

To predict whether two mentions m and its candidate antecedent n are a co-referring pair these mention representations are passed through two parallel feed-forward networks (FFNNs). One obtains scores for m and n to be a singletons separately, the second one computes a score for m and n (and all other possible antecedents of m) to be a co-referent pair. The final output is either a single mention with its score or the highest scoring mention pair. Figure 2 visualizes this inference procedure.

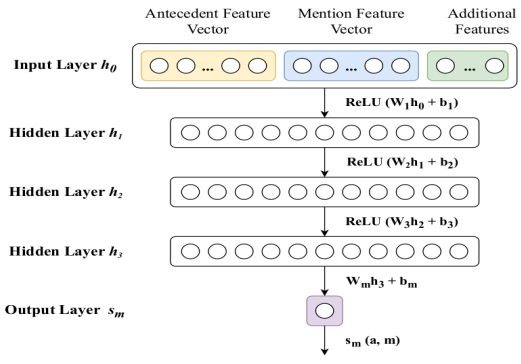

Figure 1: Neural architecture (Kupriianova et al., 2018)

Coreference chains are formed by repeating this step and finally choosing the chain which minimizes the loss.

Each FFNN consists of 3 hidden layers with ReLU activation function and dropout afterwards followed by a final scoring layer. Figure 1 describes the pass through a single one of these parallel networks, whereas 2 depicts the entire inference process with both FFNNs.

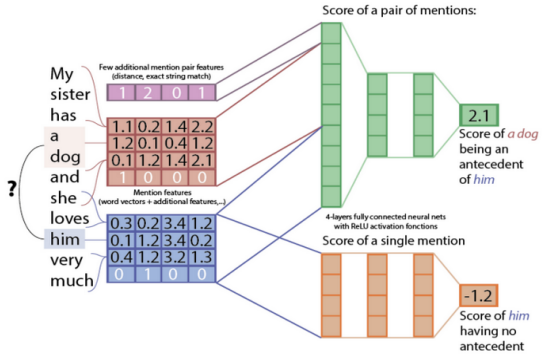

Figure 2: NeuralCoref example (Wolf, 2017)

3.2 Data pre-processing

The system requires the input data to be in the CoNLL format. In this annotation schema co-reference labels are given for all entities with at least one co-referring expression. However, this also means that singletons, which are not referenced by any other mention, are not identified during the initial mention extraction process. Therefore, one can unfortunately only evaluate the mention detection part of the system for pairs of mentions.

The mention extraction module follows a rule-based approach to identify a mention and all potential antecedents. Detected mentions are collected in a table with all preceding mentions, which they can possibly form a co-reference pair with.

In a last step the mention representations are augmented with additional features, such as span vectors and word indices, which are passed to the system at training and testing time as a separate feature vector (see Figure 2).

Before training the data is sorted and split into mini-batches according to the number of candidate antecedents for each mention. A reason for this is that mentions can have varying numbers of potential antecedents. When the mentions are sorted according to the number of antecedents to be taken into account, the maximal number of pairs P (i.e. the longest co-reference chain) to be considered is set for each batch and all mention-pairs in a batch are padded to this length.

3.3 Inference and Training

Inference and training are based on the approach proposed by (Clark and Manning, 2016a). They implement a three step training procedure with a different objective function in each iteration.

During the first two training phases the standard cross-entropy loss is optimized. The only difference is that initially all potential mention-pairs are considered (1), whereas in the second step the loss is only obtained for the top-scoring true and false antecedent candidates (2).

$$-\sum_{i=1}^{N}\left[\sum_{t\in\tau(m_i)}\log p(t,m_i)+\sum_{f\in\mathcal{F}(m_i)}\log(1-p(f,m_i))\right] \quad (1)$$

$$-\sum_{i=1}^{N}\left[\max_{t\in\tau(m_i)}\log p(t,m_i)+\min_{f\in\mathcal{F}(m_i)}\log(1-p(f,m_i))\right] \quad (2)$$

$\tau(m)$ is the set of true antecedents of a mention m, $\mathcal{F}(m)$ the false antecedents.

In the final third training phase, however, the ranking loss function is a max-margin loss. It accounts for different types of errors which the system can make because it includes an additional scoring function $\Delta(a, m_i)$.

The basic idea behind this approach comes from reinforcement learning. (Clark and Manning, 2016a) identify four *actions* which the system can perform: (*false new link* - FN, *false anaphor* - FA, *wrong link* - WL, *correct link*). The scoring function assigns a cost to each of these *actions*. It is defined as follows:

$$\Delta(a,m_i)=\begin{cases}\alpha_{FN} & \text{if } a=NA\wedge T(m_i)\neq NA\\ \alpha_{FA} & \text{if } a\neq NA\wedge T(m_i)=NA\\ \alpha_{WL} & \text{if } aNA\wedge a\notin T(m_i)\\ 0 & \text{if } a\in T(m_i)\end{cases} \quad (3)$$

The *correct* decision at one step in time then minimizes the ranking loss. Finally, the best coreference chain is the one with the smallest overall loss. The complete objective function is defined as follows:

$$\sum_{i=1}^{N}\max_{a\in A(m_i)}\Delta(a,m_i)(1+s_m(a,m_i)-s_m(\hat{t}_i,m_i)) \quad (4)$$

where $A(m_i)$ is the set of all candidate antecedents for a mention m_i and $\hat{t}_i$ the highest scoring (i.e. 'true') antecedent.

3.4 Retraining for other language

The developers of NeuralCoref provide step-by-step instructions[6] on training a new model. They claim that it is possible to train a model for other languages if the following conditions are satisfied:

- there is a corpus with coreference annotations for the language of interest (preferably in CoNLL[7] format);
- there is a model (ideally provided by spaCy) that is able to parse the language;
- there are pre-trained word vectors for the language (e.g. gloVe[8]).

However, retraining is not expected to be a direct plug-in procedure, it involves meaningful modifications of the source code such us converting the corpus to a suitable for NeuralCoref system format and adapting the mention extraction function to dependency parsing specifics of the chosen language .

In the next section, we describe an experimental retraining on the example of Russian language. Moreover, we provide evaluation for the default English model of NeuralCoref.

4 Experiments

All files containing the conducted experiments are published[9] and are accompanied by an extensive description.

4.1 Datasets

Training and test data for our experiments in Russian come from the Russian Coreference corpus

[6] https://github.com/YanaPalacheva/neuralcoref/blob/master/neuralcoref/train/training.md

[7] https://cemantix.org/conll/2012/data.html

[8] https://nlp.stanford.edu/projects/glove/

[9] https://github.com/YanaPalacheva/experiments_neuralcoref

(*RuCor*) (Toldova et al., 2014). It comprises 181 texts of different genres (news, essays, fiction, science, blog posts). In total the corpus annotates 16558 NPs and 3638 coreference chains between them.

There are several reasons why we chose the *RuCor* corpus for our experiments. First of all, the data comes in the CoNLL-style format, which supposedly can be modified with minimal effort into the required data format for training and testing within the 'NeuralCoref' pipeline.
However, only NPs referring to real-world entities and pronomial NPs are annotated as mentions. Generic or abstract nouns are counted as mentions only if they are antecedents of pronouns.

Another characteristic of the corpus which makes it particularly suitable for our purposes is the distribution of genres within the corpus. Similar to *OntoNotes 5.0*, which the original English system has been trained on, most documents are news texts. This ensures some shared basis for a future evaluation and comparison across languages.

For evaluation of the original out-of-the-box English model included in NeuralCoref and for intended comparison of Russian and English models, we use the *PAWS* corpus (Nedoluzhko et al., 2018). *PAWS* is a multi-lingual parallel treebank annotated with coreference annotation which contains English texts translated into Russian, among others. In total, there are 1,078 annotated sentences with 4,200 coreferring nodes in both languages.

The textual data is taken from the Wall Street Journal and, surprisingly, is also a part of *OntoNotes 5.0*, which, unfortunately, has not been explicitly stated in the description of *PAWS*. However, the *PAWS* corpus uses another version of coreference annotation; the differences are explained in (Nedoluzhko et al., 2016).

4.2 Retraining for Russian language

A primary direction of our work was an attempt to adapt an existing off-the-shelf system for the Russian language. While performing this experiment, we had two main objective in mind: multilinguality and reusability of existing systems.

Multilinguality is an aspect of interest because the default language in Natural Language Processing field in English and we believe it is beneficial for the whole domain to extend existing systems on other languages. The reason we picked Russian is that this language is native for two of the authors, and moreover Russian is definitely a less-researched language: there is no open and ready-to-use Coreference Resolution systems for it.

Reusability and extendability is another important issue of modern research. Those principles allow other researcher and developers to extend a published approach to new applications with a minimum of effort. This has the potential to save a great deal of time in transmitting knowledge and extend the horizon breadth-wise. It is a crucial point for Natural Language Processing sphere with its huge bias towards English language.

We are truly amazed by the work of NeuralCoref system developers. They have put a lot of effort into making their system more flexible and provided explicit training instructions along with the source code. We believe it is an important step towards extendable NLP software.

With this project, we made an attempt to train the system for Russian language. We followed the instructions kindly provided by the authors. Each step is described in further detail below.

Training steps. First of all, we have found a dataset annotated with coreferential relations: RuCor, it was described in Section 4.1. It is structured following the CoNLL format, however it was necessary to modify it further and convert it into OntoNotes-style dataset in order to minimize alterations of the system's source code. The script used to convert the corpus is published on our repository mentioned in the beginning of this section.

The next step was to make sure that spaCy supports the language. Unfortunately, spaCy published a model for Russian only recently[10] and the system is not yet adapted to the newest version. That is why we have found another parser: a spaCy wrapper of Stanza library (formerly StanfordNLP)[11], which supports Russian. It can be easily integrated into spaCy pipeline, so it required minimal intervention in the code.

The next step was to find a set of pre-trained word vectors. There are plenty of pre-trained word embedding for Russian, however the authors did not specify the format of embeddings they used for the English model. There is no documenta-

[10] https://spacy.io/usage/v3

[11] https://github.com/explosion/spacy-stanza

tion on that matter either. In their blogpost, they say they "trained our word embeddings on a large coreference annotated dataset" (Wolf, 2017), but never specified how they did it and which format one should follow. It is crucial, because there are no standards, and it is impossible for the human to infer the suitable format from files they published since word embeddings are vectors of numbers.

We tried to plug in the pre-trained word vectors from FastText, however we did not manage to debug the code and make it run since it seems like all in-line tests are strictly oriented to the dimensionality of the word embeddings used for the English model and there are no comments or other type of documentation on these numbers.

From this point, we decided to check if there are any evidence of successful adaptation for other languages, and we could not find any. It can be seen from the Issues section on the original repository[12], that developers tried to adapt it for German, French and Dutch languages. Unfortunately, many of their questions (lots of them were regarding word embeddings issue as well) are left unanswered. One of the developers who was trying to train a German language model managed to overcome this issue by loading the vectors from spaCy's German model, however the training was still unsuccessful[13].

4.3 Evaluation of NeuralCoref

Since we strongly believe in the importance of reproducibility of scientific results, a significant part of this work is dedicated to evaluation of the existing system.

The authors of NeuralCoref do not provide evaluation results for the pipeline. They, however, note that the system performance may decrease in comparison to the prototype introduced in (Clark and Manning, 2016a). They explain it by the trade-off between accuracy and speed for usability purposes, the choice of embeddings, and other smaller changes in the pipeline for universalization purposes.

We are especially interested in the model's performance on unseen data. NeuralCoref was trained on OntoNotes 5.0, so the first, preliminary, evaluation is done using the pipeline's test data obtained by following the instructions provided by the authors. Next, we feed the model the PAWS dataset to see how the performance is influenced by the choice of the annotation style and supposedly new data.

Initially, we planned to evaluate the performance of the pipeline on data similar in genre, but previously unseen by the model. However, the train and development sets apparently contain some of the documents from PAWS. Thus, the neural network has previously seen at least some of the data, although annotated with the *OntoNotes* coreference annotation style.

For evaluation, three metrics are used: MUC, B^3, and $CEAF_e$. The choice of metrics is based on the metrics used in the original (Clark and Manning, 2016a) paper, as well as on the ready-to-use implementation of the given metrics in the NeuralCoref source code. All three of the metrics are widely used in the field of coreference resolution.

As described in (Moosavi and Strube, 2016), each of the existing metric has flaws that cause under- or over-evaluation of linked entities. We include a short summary of the metrics we use along with the highlight of the main limitations.

MUC is a link-based metric. It considers the number of missing links and does not take into account what exactly was linked. Besides, the metric favors over-merged links. Thus, if a coreference system links everything together, it will obtain a higher score than in case of carefully chosen mention clusters.

B^3 is mention-based and measures how well a system detects individual mentions. The main issue is the mention identification effect: the metric does not consider whether a mention was linked to a correct entity.

Another significant flaw is connected to repeated mentions. A coreference system that links one mention to many entities would get a higher score than one that does not make hasty assumptions towards the similarity of mentions' meaning.

$CEAF_e$ measures SolarCity between two entities by looking for common mentions. In this case, mention identification problem also exists, since the metric rewards even small intersections between entities. This means that incorrectly assigned entities (possibly linked due to ambiguity or by accident) are encouraged by $CEAF_e$. The size of the compared entities is also ignored.

(Moosavi and Strube, 2016) also make notice that the $CEAF$ metric ignores correct decisions of

[12] https://github.com/huggingface/neuralcoref/issues/

[13] https://github.com/huggingface/neuralcoref/issues/257

unaligned entities. This is an important note, however, our experiments demonstrate that this problem exists for all three metrics.
Neither of the chosen metrics mark as successful the case when there are no coreference chains found both by the original annotators and the pipeline. This can only mean that the metrics do not reflect the number of correctly unmarked coreference candidates. It is a significant flow of the currently widely used coreference scores.

The evaluation results are seen in the Table 1. We compare the results of our evaluation with (Clark and Manning, 2016a). The decrease in measurements is significant even on the train data.

For partially unseen data which was annotated independently, following a slightly different annotation style, the evaluation scores hardly reach 10% accuracy. To summarize, the following are the reasons behind such a difference in performance:

- trade-off between accuracy and usability made by NeuralCoref engineers causes expected decrease in performance;
- metrics are mostly uninterpretable and measure limited aspects, which makes them less reliable;
- lack of agreement among annotators and a vast variety of different annotation systems make it harder to do an objective evaluation.

The third issue is clearly reflected by the extremely low performance on the PAWS corpus. Since both OntoNotes and PAWS contain data from the Wall Street Journal, the only significant difference between these corpora is the coreference annotation. The model used in NeuralCoref was trained to learn the annotation rules used in OntoNotes, so it is only natural that it scores low when the result is compared to another annotation style which was specifically designed as an alternative for the style used in OntoNotes.

5 Conclusion

In this project, we conducted an experiment dedicated to adaption the existing off-the-shelf coreference resolution system to Russian language. This attempt was not successful, which can be explained by lack of proper documentation, possibly our limited programming competence and time limit. We also evaluated this system on English language and compared it to the original approach it was based on. The results show that the metrics do not reflect the number of correctly unmarked coreference candidates. It is a significant flow of the currently widely used coreference scores.

References

Kevin Clark and Christopher D. Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. In *EMNLP*.

Kevin Clark and Christopher D. Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. *ArXiv*, abs/1606.01323.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1982, Seattle, Washington, USA. Association for Computational Linguistics.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking.

Jerry R. Hobbs. 1978. Resolving pronoun references. *Lingua*, 44(4):311–338.

Anna Kupriianova, Ivan Shilin, Gerhard Wohlgenannt, and Liubov Kovriguina. 2018. The implementation of the mention-ranking approach to coreference resolution in russian.

Christopher Manning. 2019. Natural language processing with deep learning - lecture 16: Coreference resolution. University Lecture.

Nafise Sadat Moosavi and Michael Strube. 2016. Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 632–642, Berlin, Germany. Association for Computational Linguistics.

Anna Nedoluzhko, Michal Novák, Silvie Cinková, Marie Mikulová, and Jiří Mírovskỳ. 2016. Coreference in prague czech-english dependency treebank. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 169–176.

Anna Nedoluzhko, Michal Novák, and Maciej Ogrodniczuk. 2018. Paws: A multi-lingual parallel treebank with anaphoric relations. In *Proceedings of the First Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 68–76.

S. Toldova, A. Roytberg, A. A. Ladygina, M. D. Vasilyeva, I. L. Azerkovich, M. Kurzukov, G. Sim, D. V. Gorshkov, A. Ivanova, A. Nedoluzhko, and Y. Grishina. 2014. Evaluating anaphora and coreference

	MUC			B^3			$CEAF_e$		
	R	**P**	**F$_1$**	**R**	**P**	**F$_1$**	**R**	**P**	**F$_1$**
(Clark and Manning, 2016b)	73.85	65.42	69.38	67.53	56.41	61.47	62.84	57.62	60.12
NeuralCoref OntoNotes	30.71	41.26	34.02	22.43	33.91	25.74	28.14	33.80	29.77
NeuralCoref PAWS	5.19	8.24	6.32	4.59	9.60	5.92	6.85	12.91	8.44

Table 1: Evaluation of NeuralCoref out-of-the-box English model. The original prototype results are compared with the results obtained on two corpora: the train corpus (*OntoNotes*) and a partially unseen corpus with different annotation style (*PAWS*).

resolution for russian. In *Komp'juternaja lingvistika i intellektual'nye tehnologii. Po materialam ezhegodnoj Mezhdunarodnoj konferencii Dialog*, pages 681–695.

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.

Sam Wiseman, Alexander M. Rush, Stuart Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1416–1426, Beijing, China. Association for Computational Linguistics.

Thomas Wolf. 2017. State-of-the-art neural coreference resolution for chatbots.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.