Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2017"

Moscow, May 31—June 3, 2017

COREFERENCE RESOLUTION IN RUSSIAN: STATE-OF-THE-ART APPROACHES APPLICATION AND EVOLVEMENT

Sysoev A. A. (sysoev@ispras.ru),
Andrianov I. A. (ivan.andrianov@ispras.ru),
Khadzhiiskaia A. Y. (sanya@ispras.ru)

Institute for System Programming of the Russian Academy of Sciences, Moscow, Russia

Coreference resolution aims at grouping textual references denoting same real world entities into clusters. Many state-of-the-art results have already been received for coreference resolution in European languages, but for Russian this area is still quite novel and underexplored. With this paper we try to fill this gap. Our article reviews existing approaches and presents their adaptation for Russian language. We carry out sufficient number of experiments to estimate efficiency of various machine learning methods and features, utilized under the hood of the algorithms. Additionally we propose a novel feature to be used for head detection subtask, which is based on word embeddings clustering. As a result, we managed to establish baseline implementation for Russian language coreference resolution problem. The key features of the developed approach are simplicity and extensibility. Presence of such a baseline opens many research directions for improving quality of the algorithms; some potential improvements are already pointed out in this paper. We expect further works in this area to significantly increase current level of state-of-the-art results for Russian coreference resolution, making it practically applicable in the near future.

Keywords: coreference resolution, anaphora resolution, mention detection, natural language processing, machine learning, word embeddings

РАЗРЕШЕНИЕ КОРЕФЕРЕНТНОСТИ: ПРИМЕНЕНИЕ И РАЗВИТИЕ СОВРЕМЕННЫХ ПОДХОДОВ

Сысоев А. А. (sysoev@ispras.ru),
Андрианов И. А. (ivan.andrianov@ispras.ru),
Хаджийская А. Ю. (sanya@ispras.ru)

Институт системного программирования
Российской академии наук, Москва, Россия

1. Introduction

Coreference resolution aims at grouping natural language expressions into clusters, according to entities of the real world, they denote. As text is naturally linear, such clusters are usually represented as chains or trees. The mention, which already has some meaning, is called antecedent; while the mention, which borrows its meaning from antecedent, is called anaphor.

There are many research projects, targeting coreference resolution problem for European languages; many state-of-the-art results have already been received. However, for Russian language this area is quite novel and the modern state can hardly be clearly defined. In this article we try to fill this gap. We review approaches commonly used to solve coreference resolution problem for foreign languages and adapt these approaches for Russian.

Thus the main contribution of the paper is threefold. First of all we apply state-of-the-art methods for coreference resolution in European languages (especially English) to Russian. We also provide detailed analysis of different algorithms and features usefulness. Additionally we present a novel feature based on word embeddings clustering for one of the building blocks—head detection algorithm.

2. Related work

Coreference resolution started its long history with initial attempts to resolve pronoun references [6]. It experienced another development in the mid 1990s, when several specific coreference resolution tasks were issued during Message Understanding Conferences. In 2001 there appeared a fundamental work [15], presenting a machine learning approach to building coreference resolution algorithms. The proposed method started from generating a number of entity mentions. Then each potential antecedent-anaphor pair was classified by pre-trained decision tree. Finally, mention pairs, classified as coreferent, were aggressively merged into clusters, each representing a separate entity.

Another epoch in coreference resolution started with ConLL-2011/2012 shared tasks for modeling unrestricted coreference, which provoked a number of novel approaches. [9] presented a multi-filter method, where each level established or forbade links between

pairs of mentions. Idea presented in [15] with mention pairs classification further evolved in [4] with more features and machine learning algorithms being used. [5,11] utilized more sophisticated approaches to coreference resolution, regarding the task as global optimization for all document mentions, in contrast to local optimization for selected mention pairs. They managed to gain state-of-the-art results on ConLL datasets.

However, [25] claimed that even simple mention-pair classification algorithms can achieve top-level results. It proposed two main improvements: easy-first mention-pair clustering algorithms, utilizing not only positive classification predictions, but also negative ones as non-grouping constraints; additionally it exploited Jaccard Item Set mining [14] feature selection to inject non-linear features into linear predictor.

Recent experiments with coreference resolution in Russian were conducted as a part of RU-EVAL 2014 evaluation campaign [17]. For its purposes the first Russian coreference corpus was compiled and manually annotated [18].

The evaluation track consisted of two tasks: coreference chains identification and anaphora resolution. Organizers reported three participants in coreference resolution task; however, they did not provide any results of the evaluation and the papers published on the matter covered mostly anaphora resolution, see [2, 7, 8, 12]. Only the first article suggested a number of rule-based techniques to resolve mentions coreferring with named entities but unfortunately they did not evaluate proposed methods.

Another research on Russian coreference was presented in [19]. This article described a machine learning based system. However, authors focused on the description of two experimental modules for sieving singleton and anaphoric mentions and did not provide detailed information about their resolver. The overall quality of their algorithms showed F1-measure of 48.04% on MUC metric [26] and 32.51% on B3 metric [1].

3. Method description

In this section we describe our method for building clusters of coreferent mentions for input text.

Our method consists of two main steps: mention detection and coreference resolution. Mention detection algorithm extracts word expressions that are possible elements of coreference chains. Extracted mentions are further grouped into coreferent clusters.

The first step of mention detection is further divided into two parts: collecting all mention heads in the given document and expanding them to full mentions. Such two-stage algorithm is inspired by [11], though the very definition of mention head in our implementation differs from the one provided in the paper.

Our algorithm requires text documents to be preprocessed, which includes morphological analysis (part of speech tags, grammemes, lemmas), syntactic analysis (dependency trees) and named entity recognition. All these steps are performed by Texterra system [20].

Additionally we train lemma-based word embeddings (skip-gram word2vec vectors with 50 dimensions) on RuEval-2014 corpus [17], FactRuEval-2016 test corpus [16], Russian section of Wikinews[1] and internal newswire corpora.

[1] https://ru.wikinews.org/

3.1. Head detection

Our system is designed to resolve entity coreference by establishing links between noun phrases and quantified phrases. Thus we consider a mention head to be a single token tagged as either noun, or numeral, or (as an exception) adjective pronoun (possessive, relative or demonstrative). We view the task of heads identification as a binary classification problem aiming to distinguish candidates as true/false mention heads. We employ a number of heuristics to obtain sure heads, which do not require further classification. Pronouns are anaphoric by definition and therefore are always incorporated in some coreferent chain. We also interpret named entities annotated by preprocessing tool as sure mentions, thus their heads are added heuristically to the resulting set. We follow named entities restriction on overlapping. All non-head nouns and numerals nested in named entities are skipped during candidate head generation step.

Our feature set for classified candidate tokens can be divided into several groups each capturing linguistic insights on various levels of language organization. Linguistic factors behind our feature set are backed up by a number of papers focused on categorizing discourse entities in light of their role in the coreferent text structure: anaphoric expressions, singletons, antecedents, etc [3, 11, 13, 23].

Internal morphological features include basic information about token such as POS-tag and corresponding grammemes: number, gender, animacy, etc. Syntax group encodes position and relations of a token within a sentence, representing candidate local salience. Syntactic context features contain morphological features for syntactic parent deduced from a dependency tree. Context group includes the same basic morphological features for two left and right token neighbours. Frequency feature set consists of tf weighting for both word form and lemma.

Semantic features utilize pre-trained word embeddings. Lemma vectors for mention heads in the training corpora are clustered by KMeans++, then cosine similarity and Euclidean distance between candidate token lemma and each of given cluster centers are exposed as features. The intuition is that each cluster represents a “sense” and high similarity (low distance) to any “sense” is highly correlated to being head. Described technique allows to attend to data sparsity problem present in most NLP tasks by recognizing heads that are not present in the training corpora but are similar to ones that are.

3.2. Head expansion

Our head expansion method is partially similar to [11]. To determine left (right) mention boundary we iterate through tokens to the left (right) starting from the nearest neighbour of mention head and apply a binary classifier until it predicts *false* label. The final token on which classifier predicts *true* label is a mention boundary. There is a simple exception for the given method: pronoun heads are treated as full mentions without any classification.

Classifier features can be categorized as following: token-based, position-based, context-based. Token-based features include word form, lemma and part-of-speech information about head/candidate token and its nearest neighbours. Position-based features include direction from head to candidate, distance between head and candidate

and whether head/candidate is the first/last token of the sentence. Context-based features reflect whether head and candidate are parts of the same named entity, whether head/candidate is a syntactic ancestor (in terms of a dependency tree) of the other one and part-of-speech pattern for words between head and candidate.

3.3. Coreference resolution algorithm

Given a collection of mentions as input coreference resolution algorithm clusters them into groups, each denoting a single entity. Our approach is highly inspired by [25]: we start from generating a number of mention pairs, which are then classified as belonging to one coreference cluster or not. Additionally, the classifier provides some confidence estimation of its decision. Finally, analyzed mention pairs are merged into clusters according to Easy-First Mention-Pair approach presented in [25].

3.3.1. Generation of Candidate Mention Pairs

Generation of candidate mention pairs occurs during two phases: training and testing.

In training the standard method [15, 25] is to construct positive examples from nearest valid coreferent pairs; each mention between members of correct pair coupled with pair's anaphor delivers negative examples.

In testing phase a window specifying a number of neighbor mentions to be taken from the left side of each text mention is usually applied. In our approach we choose the window size covering distances between mentions of 98% valid coreferent pairs within training corpus.

3.3.2. Mention pair classification

All mention pairs are converted into feature vectors, which are then classified. Populated feature vectors consist of several groups:

- **Basic linguistic features** include word forms [5, 25], lemmas, part-of-speech tags [5] and grammemes (gender, number, animacy) [4] for mention head and context words. Context is composed from up to two same sentence tokens to the left and to the right of the considered mention [5, 25].
- **Grammemes agreement features** are indicators of mention heads sharing the same key grammemes (number [15, 24, 25], gender [4], animacy [24, 25], pronominality [5]).
- **Positional features** provide information about mentions arrangement in text: distance [5,15] and place within sentence boundaries [24,25].
- **Named entity based features** provide information about mention types and their agreement [5, 24, 25].
- **Structural features** encode information about mention size [5, 25] and interrelation with other mentions of the text (intersecting [22]; containing other mentions or being contained in other mentions [5, 24, 25]).
- **Surface form matching features** include lexicographic similarity [21] and textual representation equality indicators [5, 15, 24, 25]. In our approach features of this group are based on lemmas, constituting mentions, rather than on their word forms.
- **Syntactic features** incorporate information, that can be extracted from dependency trees, such as grammar role, sharing same parent node or clause and so on [22].

Additionally, conforming to [25], in conjunction with linear classifier we use Jaccard Item Set mining algorithm [14] to gain sets of features, frequently appearing together. Joining these features into a single composite provides the ability to utilize even primitive features, targeting only one mention of the classified pair.

3.3.3. Easy-First Mention-Pair algorithm

Easy-First Mention-Pair algorithm [25] receives a number of candidate mention pairs together with their classification results—whether the pair is valid or not and confidence of this prediction. Provided pairs are sorted by confidence, so that more precisely classified pairs come first. Initially, each mention is assigned to its own cluster. Confidence-ordered list of mention pairs is walked down sequentially: pairs, classified as valid, are merged into a single cluster; pairs, classified as invalid, are memorized as unlinking constraint to prevent merging further pairs with lower confidence. If merging two clusters results in one containing a previously unlinked pair the analyzed pair is just ignored. Clusters present after full list traversal represent target groups of coreferent mentions.

4. Evaluation

In this section we describe corpus used for testing and present detailed evaluation results for each of the algorithm steps.

4.1. RuEval-2014 corpus

We employ the RuCor coreference corpus [17] as a training and evaluation set for all subtasks described above. It contains 181 texts (about 200,000 tokens) representing five written genres: news, essays, fiction, scientific articles and blog posts.

During our experiments we encountered two major problems in the original dataset that required manual fixes: duplicated mentions and cyclic chains. The first problem was caused by erroneous merging of markup variants from different annotators. In order to avoid merging discrepancies we took into consideration only mentions with the most popular variant label “1”. Two documents from the original dataset contained no variants with this label and had to be dismissed. We also detected and manually straightened 6 chains containing cycles.

Another problem with the dataset arose from different definitions of mention head. Some tokens that could serve as a head for potential mention were not annotated. Sometimes for the purposes of evaluation campaign annotators marked several heads in dubious cases, such as appositional proper names and coordinate noun phrases, 1,696 multi-token heads in total. Following the constraints of dependency parsing our system expects head to be a single token. In order to process these cases we applied simple heuristic taking the first noun as a correct head.

These inconsistencies make it difficult to evaluate absolute quality of the algorithms with RuCor dataset, but comparative analysis appears reasonable.

4.2. Head detection

We employ groundtruth heads retrieved from corpus to test the quality of our head detection algorithm in 10-fold cross-validation applying standard metrics such as precision, recall and F1-measure. We invoke our detector with and without semantic features (based on word embeddings) to evaluate impact of this feature group on the overall quality and make several runs to find the optimal number of clusters (Table 1).

Table 1. Head detection evaluation

Setting: logistic regression with L2 regularization	**Precision**	**Recall**	**F1-measure**
without semantic features	**0.7326**	0.6628	0.6952
with 105 clusters	0.7289	0.6839	0.7050
with 110 clusters	0.7288	0.6841	0.7051
with 115 clusters	0.7289	**0.6842**	**0.7052**
with 120 clusters	0.7280	0.6839	0.7046
with 125 clusters	0.7288	0.6838	0.7049

Table 1 shows that semantic features are highly influential and can dramatically increase recall and F1-measure. The best general quality is reached with 115 clusters therefore we use this number in the full pipeline as an empirically preset variable.

4.3. Head expansion

Given gold mention heads we evaluate our head expansion algorithm in 10-fold cross-validation with three information retrieval metrics: precision, recall, F1-measure. We additionally provide ablation analysis results for context-based features, as their usefulness is most controversial (Table 2).

Table 2. Head expansion evaluation

Setting: logistic regression with L2 regularization	**Precision**	**Recall**	**F1-measure**
all	0.8682	0.8654	0.8668
all—same NE	0.8652	0.8627	0.8640
all—syntactic ancestor	0.8483	0.8440	0.8461
all—POS pattern	**0.8687**	**0.8660**	**0.8673**
all—context features	0.8441	0.8401	0.8421

As we can see the most influential context-based feature is syntactic ancestor. This looks reasonable as mentions are generally expected to be subtrees of the sentence syntactic tree. The worst context-based feature is POS pattern, it even introduces minor loss in all metrics, which requires additional analysis.

4.4. Coreference resolution

MUC [26], B3 [1], $CEAF_{entity}$ and $CEAF_{mention}$ [10] versions of precision, recall and F1-measure are used to evaluate performance of coreference resolution approach in 10-fold cross-validation. Experiments within this section are carried out with ground-truth mentions, thus $CEAF_{mention}$ metrics are all the same.

First of all, we examine impact of different classification algorithms utilized under the hood of coreference resolution (Table 3).

Table 3. Evaluation of machine learning algorithms in coreference resolution

Metric \ Machine learning algorithm		**logistic regression**	**logistic regression + Jaccard Item Set mining**	**random forest**
MUC	Precision	0.7246	0.7333	**0.7395**
	Recall	0.6969	**0.7027**	0.6520
	F1	0.7104	**0.7175**	0.6928
B3	Precision	0.5852	0.6014	**0.7389**
	Recall	**0.6104**	0.6103	0.5516
	F1	0.5973	0.6055	**0.6312**
$CEAF_{mention}$	Precision/Recall/F1	0.5284	0.5375	**0.5894**
$CEAF_{entity}$	Precision	0.4765	**0.4825**	0.4780
	Recall	0.5396	0.5514	**0.6583**
	F1	0.5057	0.5140	**0.5533**

As it can easily be seen, Jaccard Item Set mining algorithm, introducing non-linear features into logistic regression, slightly improves its quality. However, random forest manages to show even better progress. This result looks reasonable, as decision trees within random forest algorithm are naturally designed to induce knowledge from combinations of even most trivial features. Jaccard Item Set mining should have performed similarly, however it is probable that we did not succeed to choose optimal parameters for it.

To determine most significant features for coreference resolution task ablation analysis experiments (with random forest classifier) are carried out (Fig. 1, 2).

Basic linguistic and syntactic features seem useless for coreference resolution tasks—removing them might even slightly improve results, while surface form features are the most valuable ones.

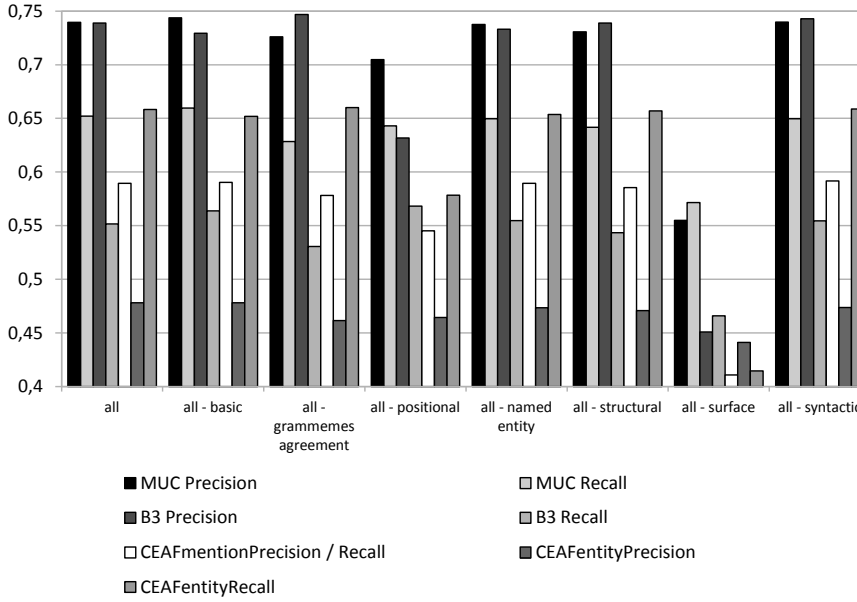

Fig. 1. Ablation analysis for coreference resolution. Precision and recall

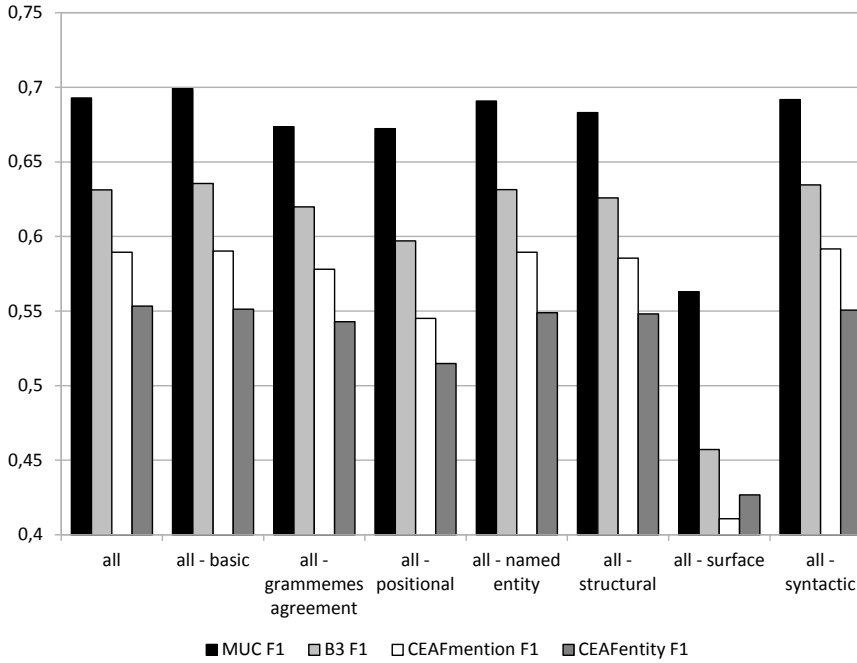

Fig. 2. Ablation analysis for coreference resolution. F1

4.5. Full coreference resolution pipeline

This section presents the results of evaluating coreference pipeline as a whole—starting from head detection and expansion. During each iteration of 10-fold cross-validation parts of the algorithm were sequentially trained and then applied to testing documents. Final results are presented in Table 4.

Table 4. Evaluation results of full coreference resolution pipeline

Precision	Recall	F1	Precision	Recall	F1
MUC			B3		
0.4768	0.3741	0.4189	0.4104	0.2957	0.3431
$CEAF_{mention}$			$CEAF_{entity}$		
0.4024	0.3702	0.3854	0.2525	0.3433	0.2906

These results significantly differ from the scenario with ground-truth mentions which is explained with accumulation of errors of all intermediate steps.

5. Conclusion

In this paper we aimed to evaluate the usefulness of coreference resolution approaches, developed for European languages, when applied to Russian. Presumably, we accomplished some baseline implementation for this problem. The key features of the developed approach are simplicity and extensibility, which opens many research lines in this area. We consider the following directions to be beneficial in the near future:

- carrying out experiments with more machine learning algorithms and approaches;
- using various clustering algorithms for word embeddings;
- detailed analysis of features, assumed useless in ablation experiments;
- tuning coreference resolution algorithm for different mention types.

References

1. *Bagga A., Baldwin B.* (1998), Algorithms for Scoring Coreference Chains, The first international conference on language resources and evaluation workshop on linguistics coreference, pp. 563–566.
2. *Bogdanov, A. V., et al.* (2014), Anaphora analysis based on ABBYY Compreno linguistic technologies, Computational Linguistics and Intellectual Technologies, issue 13, pp. 89–102.
3. *De Marneffe, M.-C., Recasens M., Potts C.* (2015), Modeling the lifespan of discourse entities with application to coreference resolution, Journal of Artificial Intelligence Research, iss. 52, pp. 445–475.
4. *Dos Santos C. N., Carvalho D. L.* (2011), Rule and Tree Ensembles for Unrestricted Coreference Resolution, Proceedings of the Fifteenth Conference on Computational Natural Language Learning, pp. 51–55.

5. *Fernandes E. R., Dos Santos C. N., Milidiú R. L.* (2012), Latent Structure Perceptron with Feature Induction for Unrestricted Coreference Resolution, Joint Conference on EMNLP and CoNLL-Shared Task, pp. 41–48.
6. *Hobbs J.* (1978), Resolving pronoun references, Lingua, vol. 44, issue 4, pp. 311–338.
7. *Ionov M., Kutuzov, A.* (2014), The impact of morphology processing quality on automated anaphora resolution for Russian, Computational Linguistics and Intellectual Technologies, issue 13, pp. 232–240.
8. *Kamenskaya M. A., Khramoin I. V., Smirnov I. V.* (2014), Data-driven methods for anaphora resolution of Russian texts, Computational Linguistics and Intellectual Technologies, issue 13, pp. 241–250.
9. *Lee H., Peirsman Y., Chang A., Chambers N., Surdeanu M., Jurafsky D.* (2011), Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task, Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, pp. 28–34.
10. *Luo, X.* (2005), On Coreference Resolution Performance Metrics, Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 25–32.
11. *Peng H., Chang K.-W., Roth D.* (2015), A Joint Framework for Coreference Resolution and Mention Head Detection, CoNLL, vol. 51, pp. 12–22.
12. *Protopopova E. V., Bodrova A. A., Volskaya S. A., Krylova I. V., Chuchunkov A. S., Alexeeva S. V., Bocharov V. V., Granovsky D. V.* (2014), Anaphoric Annotation and Corpus-based Anaphora Resolution: an experiment, Computational Linguistics and Intellectual Technologies, issue 13, pp. 562–571.
13. *Recasens M., De Marneffe M.-C., Potts C.* (2013), The Life and Death of Discourse Entities: Identifying Singleton Mentions, HLT-NAACL, pp. 627–633.
14. *Segond M., Borgelt C.* (2011), Item Set Mining Based on Cover Similarity, Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 493–505.
15. *Soon W. M., Ng H. T., Lim D. C. Y.* (2001), A Machine Learning Approach to Coreference Resolution of Noun Phrases, Computational linguistics, vol. 27, issue 4, pp. 521–544.
16. *Starostin A. S. et al.* (2016), FactRuEval 2016: Evaluation of Named Entity Recognition and Fact Extraction Systems for Russian, Computational Linguistics and Intellectual Technologies, issue 15, pp. 702–720.
17. *Toldova S. et al.* (2014), RU-EVAL-2014: Evaluating anaphora and coreference resolution for Russian, Computational Linguistics and Intellectual Technologies, issue 13, pp. 681–694.
18. *Toldova S., Ghrishina Y., Ladygina A., Vasilyeva M., Sim G., Azerkovich I.* (2016), Russian Coreference Corpus, Input a Word, Analyze the World: Selected Approaches to Corpus Linguistics, Cambridge Scholars Publishing, Newcastle Upon Tyne, pp. 107–124.
19. *Toldova S., Ionov M.* (2016), Mention Detection for Improving Coreference Resolution in Russian Texts: A Machine Learning Approach, Computación y Sistemas, vol. 20, issue 4, pp. 681–696.
20. *Turdakov D. Y. et al.* (2014), Texterra: A framework for text analysis, Programming and Computer Software, vol. 40, issue 5, pp. 288–295.

21. *Uryupina O.* (2004), Evaluating Name-Matching for Coreference Resolution, Proceedings of LREC, pp. 1339–1342.
22. *Uryupina O.* (2006), Coreference Resolution with and without Linguistic Knowledge, Proceedings of LREC, pp. 893–898.
23. *Uryupina O.* (2009), Detecting anaphoricity and antecedenthood for coreference resolution, Procesamiento del lenguaje natural, issue 42, pp. 113–120.
24. *Uryupina O., Moschitti A., Poesio M.* (2012) BART goes multilingual: The UniTN / Essex submission to the CoNLL-2012 Shared Task, Joint Conference on EMNLP and CoNLL-Shared Task, pp. 122–128.
25. *Uryupina O., Moschitti A.* (2015), A State-of-the-Art Mention-Pair Model for Coreference Resolution, Lexical and Computational Semantics (SEM 2015), pp. 289–298.
26. *Vilain M., Burger J., Aberdeen J., Connolly D., Hirschman L.* (1995), A model-theoretic coreference scoring scheme, Proceedings of the 6th conference on Message understanding, pp. 45–52.