Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference “Dialogue 2018”

Moscow, May 30—June 2, 2018

FRAMEWORK FOR RUSSIAN PLAGIARISM DETECTION USING SENTENCE EMBEDDING SIMILARITY AND NEGATIVE SAMPLING

Belyy A. V. (anton.belyy@gmail.com)[1,2],
Dubova M. A. (marina.dubova.97@gmail.com)[3]

[1]ITMO University, Saint Petersburg, Russia;
[2]B Tochka Bank QIWI Bank (JSC), Yekaterinburg, Russia;
[3]Saint Petersburg State University, Saint Petersburg, Russia

In this paper, we propose a new approach for advanced plagiarism detection in Russian language. It is based on a classifier, dealing with two different types of sentence similarity measures: token set similarity and cosine similarity between sentence embeddings (based on pre-trained RusVectōrēs, unsupervised fastText, and supervised StarSpace models). The diversity of feature space makes it possible to detect different types of plagiarism, starting from simple copy&paste cases and ending with complex manual paraphrases. The proposed approach implies an ability to focus on the particular plagiarism type identification, allowing to train a universal model at the same time. The method shows great results on detection of different types of plagiarism and outperforms the previous approach.

Keywords: plagiarism detection, sentence similarity, word embeddings, negative sampling

МЕТОД ПОИСКА ТЕКСТОВЫХ ЗАИМСТВОВАНИЙ НА ОСНОВЕ ВЕКТОРНОЙ БЛИЗОСТИ ПРЕДЛОЖЕНИЙ С ОТБОРОМ КАНДИДАТОВ

Белый А. В. (anton.belyy@gmail.com)[1,2],
Дубова М. А. (marina.dubova.97@gmail.com)[3]

[1]Университет ИТМО, Санкт-Петербург, Россия;
[2]Ф Точка Банк КИВИ Банк (АО), Екатеринбург, Россия;
[3]Санкт-Петербургский Государственный Университет, Санкт-Петербург, Россия

1. Introduction

Plagiarism is a perennial and pervasive problem in research and educational institutions [Maurer et al., 2006]. Nowadays, it is becoming even more widespread as a number of free Internet resources grows. Failures to appropriately acknowledge original sources discount a reward system in science [Resnik et al., 2005] and obstruct scientific search of the relevant literature. In education, plagiarism spoils an assessment process [Larkham and Manns, 2002] and makes it difficult to guarantee a correspondence between certificates and real knowledge. Thus, it impairs science and education systems and turns out to be one of the most serious problems in these fields.

In Russia, the research on advanced plagiarism detection methods is only starting to gain popularity. In 2016–2017, the first competition, PlagEvalRus-2017, was held. Organizers provided participants with a large and diverse dataset in Russian [Smirnov et al., 2017], [Sochenkov et al., 2017], and made an appropriate evaluation possible [Smirnov et al., 2017].

It this paper, we propose a new solution to the Russian plagiarism detection problem. Our approach employs the diversity of Natural Language Processing methods to deal with plagiarism cases of different complexity. It achieves great results and outperforms the previous method in detecting plagiarism of all types.

2. Task and dataset

2.1. Task

Scientists split the problem of plagiarism detection into two main parts: source retrieval and text alignment (for a review of subtasks, see [Alzahrani et al., 2012]). Source retrieval is a search of all possible sources for documents suspected to borrow information, whereas text alignment is a search of particular contiguous passages of text borrowed from a given source. In our work, we consider only the second task and address an extrinsic plagiarism detection. It means that we already have suspicious documents with source candidates and try to find particular fragments, matching in suspected and source documents.

2.2. Dataset

There are plenty of datasets for plagiarism text alignment in English, which were employed at the special PAN competitions held annually from 2009 to 2014. The Russian dataset is a new one (for a detailed description, see Smirnov et al., 2017, Sochenkov et al., 2017), nevertheless, it was developed according to the worked-out “gold-standards” of English plagiarism detection datasets. It consists of three main types of plagiarism:

- Automatically generated copy&paste plagiarism (copy&paste)
- Automatically generated paraphrased plagiarism (paraphrased)
- Manually generated paraphrased plagiarism (manual)

3. Related work

The main contributions to the plagiarism text alignment were made at the PAN annual competitions, and the latest winning approach is described in [Sanchez-Perez et al., 2014]. In this method, authors use TF-IDF weighted vectors of sentences and cosine similarities and the Dice coefficients between them for comparison. Afterwards, scientists usually focused their work on the particular type of plagiarism (in most of cases, the manual type) and achieved significant improvements by applying genetic algorithms [Vani and Gupta, 2017]; [Sanchez-Perez et al., 2017], word embedding models [Brlek et al., 2016], and topic modeling methods [Le et al., 2016].

However, only one method is adapted to Russian language. This algorithm [Zubarev and Sochenkov, 2017] was developed in accordance with a specificity of Russian language and PlagEvalRus-2017 dataset. Authors applied semantic and syntactic parsing to compute textual similarity score, used to detect plagiarism cases. This method was developed and evaluated at the first competition on advanced plagiarism detection methods in Russia.

4. Method

4.1. Preprocessing

4.1.1. Text splitting

We split each document into sentences using *nltk* library. Afterwards, each sentence is tokenized using simple regular expression and lemmatized using *mystem* library to reduce the size of the vocabulary. To account for homographs (words with the same spelling, but different pronunciation and meaning, like добро́$^{\text{noun}}$, добро́$^{\text{part}}$ and до́бро$^{\text{adv}}$ in Russian), we also store part of speech (PoS) tags of each lemma estimated using *mystem*. Sentences of short length (less than three tokens) are merged together with adjacent longer sentences to reduce granularity of predictions. After this step, we obtain a list of sentences, which are represented as sequences of lemmas with PoS tags.

4.1.2. Sentence matching

For text classifier training set (see 4.2.1) we need to construct sentence-to-sentence matchings from text-to-text matchings originally provided in the dataset. In the most common case, we have a short text of n sentences from a suspicious document which we want to match with a short text of m sentences from a source document and obtain $f(m,n)$ sentence-to-sentence pairs. In our work we experimented with two simple types of matching:

1. parallel matching: each sentence i from the suspicious text is matched with the corresponding sentence i from the source text, producing $min\,(m,n)$ matchings
2. pairwise matching: each sentence from the suspicious document is matched with all sentences from the source text, producing $m * n$ matchings

Each matching makes sense for different corpora, depending on the distribution of sentence lengths from the particular dataset. For PlagEvalRus-2017 dataset, we use parallel matching in copy&paste and paraphrased datasets and use pairwise matching in manual datasets.

4.2. Model

4.2.1. Classifier

A key component of our method is a classifier, predicting a fact of plagiarism for a given pair of sentences. In experimental phase, we tested and compared three classifiers:

- Logistic Regression (with L2 regularization and C=1.0)
- Random Forest (with 10 trees)
- Bayesian classifier, where a posteriori probabilities $p(y|x)$ are obtained using non-parametric kernel density estimation (KDE) described in [O'Brien et al, 2016].

4.2.2. Feature space

To obtain the utility of our approach for very different types of plagiarism, we decided to combine simple similarity measures along with more complex and modern ones, such as cosine similarity of sentence embeddings. Each pair of sentences (u,v) is therefore represented as a sequence of different similarity measures: $(s_1(u,v), s_2(u,v), \ldots, s_n(u,v))$, where each measure $s_i(u,v)$ is normalized to have value range $[0;1]$ and $s_i(u,u) = 1$.

4.2.2.1 Token similarity

Given tokens of a pair of sentences, denoted as u and v, we calculate how much common vocabulary do these two sentences share, using the following measures:

- $LeftInclusion(u,v) = \frac{|u \cap v|}{|u|}$
- $RightInclusion(u,v) = \frac{|u \cap v|}{|v|}$

A combination of these measures can confidently detect a copy&paste or a light paraphrase, however, it performs poorer in harder cases, for which we need to measure sentence similarity in a different way.

4.2.2.2 Sentence embeddings

Given a pair of sentences u and v, represented as sequences of tokens, we would like to train a mapping (or embedding) f from sentence to vector, such that if sentences u and v are semantically related, then the angle between vectors $f(u)$ and $f(v)$ is close to 0 (or, equivalently, cosine distance of $f(u)$ and $f(v)$ is close to 1):

$$\frac{< f(u), f(v) >}{|f(u)| \cdot |f(v)|} \approx 1$$

Our approach is based on combining several kinds of word embeddings, each optimized for different objective and therefore capturing different aspects of sentence similarity:

- RusVectōrēs [Kutuzov and Kuzmenko, 2017] pre-trained model, which was trained on Russian National Corpus and Russian Wikipedia (600 million tokens, resulting in 392,000 unique word embeddings), thus allowing to introduce more general similarities into the model.
- fastText [Joulin et al., 2016] unsupervised model, which was trained on texts from PlagEvalRus-2017 datasets (12 million tokens, resulting in 184,000 unique word embeddings). One useful feature of the fastText model is that it uses internally character N-grams rather than tokens, which is helpful in the presence of rare / out-of-vocabulary words.
- StarSpace [Wu et al., 2017] supervised model, which was also trained on PlagEvalRus-2017 corpus (resulting in 97,000 unique word embeddings). This recently introduced framework has many use-case scenarios, one which is supervised similarity learning between sentences.

We obtain sentence embeddings by averaging embeddings of individual words multiplied by their IDF weights estimated on each dataset separately [Ferrero et al., 2017]:

$$f(sent) = \frac{1}{|sent|} \sum_{w \in sent} f(w) \cdot idf(w)$$

4.3. Negative sampling

Data points in the original dataset are coordinates of textual fragments in source and suspicious documents, which correspond to each positive plagiarism case. The resting pairs of short texts can be used as negative cases to train the classifier. However, this approach has two important disadvantages: high imbalance of classes in the training set (as reflected in Table 1) and time consumption. This forced us to try **Random-*N*** negative sampling. In this approach, *N* negative examples per every positive are chosen randomly from the whole set of negative examples at the train time.

Table 1. Label distribution in different parts of PlagEvalRus-2017 dataset

	Plagiarism	Non-plagiarism	All	Plagiarism, %
Copy&paste	95,122	101,796,105	101,891,227	0.09%
Paraphrased	122,867	113,941,646	114,064,513	0.11%
Manual	18,366	16,856,074	16,874,440	0.11%
Manual2	4,241	2,648,296	2,652,537	0.16%

4.4. Granularity reduction

We reformulate the original text alignment problem to binary classification over pairs of sentences. After that, we need to merge adjacent plagiarised sentences into contiguous passages to reduce granularity (section 6.1) of predictions. We propose a simple algorithm that runs in linear time with respect to the number of sentence-to-sentence detections:

Algorithm 1. Granularity reduction
Parameters:

- gap_{susp}, gap_{src}—maximum distance between adjacent sentences from suspicious and source documents;

Variables:

- D—list of all detections in a pair of documents, ordered by suspicious sentence id;
- S—mapping from detection to a component id;

```
1. for each detection(susp_i,src_j) ∈ D:
2.     if(susp_i,src_j):
3.         S[susp_i,src_j]= new component id
4.     for i in 0..gap_susp:
5.         for j in 0..gap_src:
6.             if (susp_i+i,src_j+j) ∈ D:
7.                 S[susp_i+i,src_j+j]=S[susp_i,src_j]
```

After that, we merge detections with the same component id into a single prediction. Parameters gap_{susp}, gap_{src} are the maximum look-ahead distance of the algorithm and need to be tuned for each dataset.

5. Model selection and parameter tuning

5.1. Feature space

Feature space size put some limitations on the set of possible classifiers and account for the time and memory consumption. Considering these problems, we reduced the initial feature space from 5 to 2 measures by grouping and averaging metrics of the similar nature:

- TokenSim = ½ (LeftInclusion + RightInclusion)
- SentEmbed = ⅓ (RusVectōrēs + StarSpace + fastText)

We will use manually paraphrased part of PlagEvalRus-2017 dataset to compare classifiers in the rest of this section.

As captured in Table 2, performances of LR and RF did not change significantly. Moreover, the reduction to 2 features makes our results more interpretable and susceptible for further analysis.

Table 2. Train results for plagiarism detection on the manually paraphrased plagiarism

	Granularity	Macro			Micro		
		Precision	Recall	Plagdet	Precision	Recall	Plagdet
PAN Baseline	1.2145	**0.8899**	0.4471	0.5189	**0.9393**	0.5419	0.5992
Log Regression 2D	**1.0001**	0.7246	0.8558	0.7848	0.8173	0.8759	0.8455
Log Regression 5D	**1.0001**	0.7300	0.8537	0.7870	0.8176	0.8733	0.8445
Random Forest 2D	1.0003	0.7809	0.9340	**0.8504**	0.8347	0.9450	0.8862
Random Forest 5D	1.0004	0.7690	**0.9487**	0.8492	0.8263	**0.9566**	**0.8865**

5.2. Negative sampling

We compare sampling methods to see how they affect train time and quality of sentence similarity prediction task. For this comparison, we took 20% sentence pairs as a hold-out set and used the rest 80% for training. A quality measure is classifier's ROC-AUC on hold-out pairs.

Table 3. Quality of sentence similarity on 2D features with undersampling

	No sampling	Random-1	Random-10	Random-100
Log Regression	**0.9785**	**0.9785**	**0.9785**	**0.9785**
Random Forest	0.9322	0.9734	0.9669	0.9533
Bayesian KDE	0.9755	0.9762	0.9757	0.9758

Table 4. Training time for sentence similarity on 2D features with undersampling

	No sampling	Random-1	Random-10	Random-100
Log Regression	**22.45**	**0.06**	**0.21**	**2.27**
Random Forest	383.70	0.24	1.76	26.10
Bayesian KDE	40.47	2.16	2.63	6.28

We see that Logistic Regression (LR) achieves the best quality in the smallest amount of time. Bayesian KDE is on par with LR but takes more time to train. Random Forest demonstrates the worst quality and training time. Importantly, all classifiers show their best performance in Random-1 sampling, which justifies the use of undersampling in plagiarism detection tasks.

However, we should still beware class imbalance in test time. Below are decision boundaries for LR-2D classifier trained with different sampling parameters:

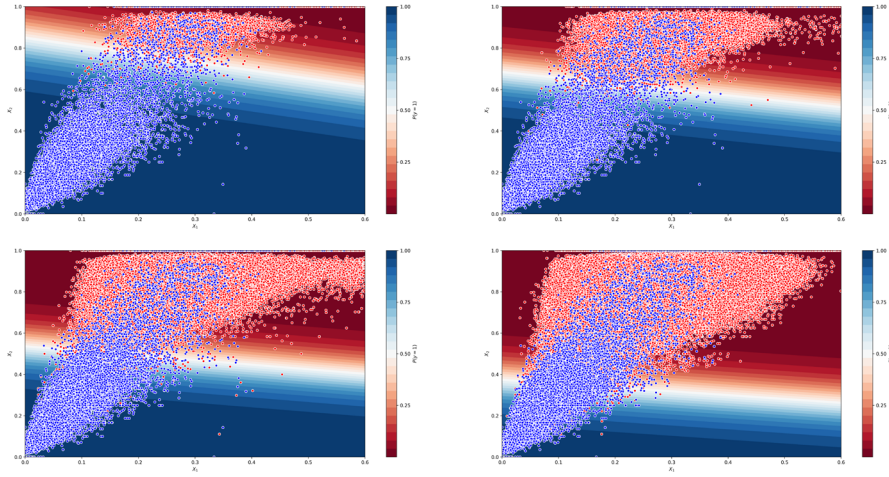

Figure 1. LR-2D trained on Random-*N* for *N* = 1, 10, 100 and no sampling

As the number of negative samples grows, the slope of decision line remains roughly the same, but its margin (*y*-intercept) gets closer to zero. In order to achieve good quality on test set, we have to tune classifier's margin b on non-sampled labeled hold-out set $S = (Sx, Sy)$ using some function Q which measures how close are predictions of trained classifier $a(Sx)$ to true labels Sy:

$$b = argmax_b Q(Sy, [a(Sx) - b > 0])$$

For Q we propose to use either classifier F_1-score or train Plagdet score (defined in section 6.1). The former takes less time to compute, but the latter is a more accurate estimate of test Plagdet score. Below are results for tuning b with classifier F_1-score:

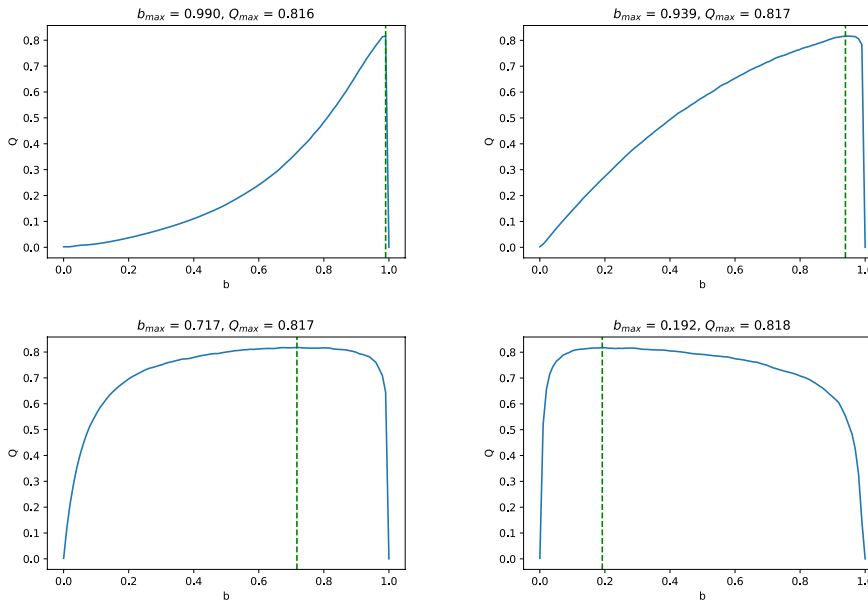

Figure 2. Tuning margin for LR-2D with Random-N for N = 1, 10, 100 and no sampling

5.3. Parameter tuning

For the whole model we need to tune three hyperparameters: gap_{susp}, gap_{src} (section 4.4) and margin b (section 5.2) and choose N for Random-N. To compute test Plagdet, we selected the best (in terms of ROC-AUC) classifier from section 5.2—LR-2D with Random-1 sampling. We trained four distinct models (three on different parts of the dataset and one on the whole dataset) and selected their hyperparameters using grid search on a linear and logarithmic scale to maximize train Plagdet:

Table 5. Hyperparameters for models

	Random-N	Granularity gap_{susp}	Granularity gap_{src}	Margin b
Copy&paste	1	10	10	$1-10^{-6}$
Paraphrased	1	10	10	$1-10^{-5}$
Manual	1	0	0	$1-10^{-2}$
All parts	1	10	10	$1-2\cdot10^{-5}$

6. Results

6.1. Evaluation metrics

Standard evaluation metrics for text alignment task are [Potthast et al., 2010]:

- Macro-averaged and micro-averaged **precision** and **recall**.

$$prec_{macro}(S,R) = \frac{1}{|R|}\sum_{r\in R}\frac{|\cup_{s\in S}(s\sqcap r)|}{|r|} \qquad prec_{micro}(S,R) = \frac{|\cup_{(s,r)\in(S\times R)}(s\sqcap r)|}{|\cup_{r\in R} r|}$$

$$rec_{macro}(S,R) = \frac{1}{|S|}\sum_{s\in S}\frac{|\cup_{r\in R}(s\sqcap r)|}{|s|} \qquad rec_{micro}(S,R) = \frac{|\cup_{(s,r)\in(S\times R)}(s\sqcap r)|}{|\cup_{s\in S} s|}$$

- **Granularity** measure. It represents a mean number of plagiarism detections per a single positive case. Ideally, an algorithm should find only one case of plagiarism for each fragment.

$$gran(S,R) = \frac{1}{|S_R|}\sum_{s\in S_R}|R_S|$$

- **Plagdet** score: a combination of precision, recall, and granularity, considering the importance of each measure. It is considered to be the most representative and valid metric for evaluation of plagiarism detection methods.

$$plagdet(S,R) = \frac{F_1}{log_2(1+gran(S,R))},$$

where F_1 is a harmonic mean of macro- or micro-averaged precision and recall.

6.2. Testing

We trained LR-2D model with Random-1 sampling on different parts of the dataset and came up with four different models: trained on all, copy&paste, paraphrased and manual types of plagiarism respectively. Afterwards, we evaluated them on hold-out test sets from PlagEvalRus-2017 corpora (for evaluation details, see Smirnov et al., 2017). Models were tested on certain types of plagiarism and on the whole dataset, obtained results are presented in Tables 1–4.

Table 6. Test results for copy&paste plagiarism

	Granularity	Macro			Micro		
		Precision	Recall	Plagdet	Precision	Recall	Plagdet
PAN Baseline	**1.0046**	0.7240	0.9101	0.8038	0.9615	**0.9943**	**0.9744**
zubarev17.1	1.5084	**0.9496**	0.6427	0.5778	**0.9828**	0.8217	0.6746
zubarev17.2	1.4660	0.9320	0.7013	0.6146	0.9776	0.8588	0.7022
Belyy: all types	1.0202	0.8168	0.8790	0.8347	0.9131	0.9717	0.9280
Belyy: copy&paste	1.0066	0.8711	0.8269	**0.8444**	0.9447	0.9396	0.9377

	Granularity	Macro			Micro		
		Precision	Recall	Plagdet	Precision	Recall	Plagdet
Belyy: paraphrased	1.0147	0.8297	0.8718	0.8413	0.9210	0.9665	0.9333
Belyy: manual	1.1336	0.3652	**0.9322**	0.4800	0.5994	0.9935	0.6839

Table 7. Test results for paraphrased plagiarism

	Granularity	Macro			Micro		
		Precision	Recall	Plagdet	Precision	Recall	Plagdet
PAN Baseline	3.4639	0.9051	0.6895	0.3626	0.9710	0.8334	0.4156
zubarev17.1	1.5404	**0.9604**	0.6730	0.5884	**0.9875**	0.8219	0.6670
zubarev17.2	1.4834	0.9473	0.7340	0.6303	0.9812	0.8650	0.7006
Belyy: all types	1.0111	0.8535	0.8788	0.8591	0.9186	0.9649	0.9337
Belyy: copy&paste	**1.0039**	0.9169	0.7760	0.8382	0.9532	0.9113	0.9292
Belyy: paraphrased	1.0074	0.8694	0.8668	**0.8635**	0.9286	0.9579	**0.9380**
Belyy: manual	1.1378	0.4259	**0.9460**	0.5359	0.6179	**0.9936**	0.6951

Table 8. Test results for manual plagiarism

	Granularity	Macro			Micro		
		Precision	Recall	Plagdet	Precision	Recall	Plagdet
PAN Baseline	1.1414	0.8332	0.0554	0.0946	**0.8960**	0.0761	0.1277
zubarev17.1	1.0015	0.8068	0.3409	0.4788	0.8845	0.3815	0.5325
zubarev17.2	1.0016	0.6250	0.4715	**0.5369**	0.8208	0.5312	0.6443
Belyy: all types	1.0000	0.8054	0.2824	0.4181	0.7910	0.2993	0.4343
Belyy: copy&paste	1.0000	0.8384	0.0505	0.0953	0.8714	0.0539	0.1015
Belyy: paraphrased	1.0000	**0.8500**	0.2138	0.3417	0.8208	0.2223	0.3499
Belyy: manual	1.0038	0.2412	**0.8700**	0.3767	0.6030	**0.8912**	**0.7173**

Table 9. Test results for all types of plagiarism

	Granularity	Macro			Micro		
		Precision	Recall	Plagdet	Precision	Recall	Plagdet
PAN Baseline	1.9953	0.8525	0.3366	0.3049	0.9637	0.6893	0.5078
zubarev17.1	1.3028	**0.9129**	0.4605	0.5087	**0.9693**	0.7043	0.6780
zubarev17.2	1.2417	0.8158	0.5644	0.5729	0.9460	0.7737	0.7309
Belyy: all types	1.0104	0.8312	0.5075	**0.6256**	0.9015	0.7898	**0.8357**
Belyy: copy&paste	**1.0048**	0.8910	0.3340	0.4842	0.9472	0.6928	0.7975
Belyy: paraphrased	1.0080	0.8492	0.4612	0.5944	0.9158	0.7648	0.8288
Belyy: manual	1.0564	0.3506	**0.8961**	0.4846	0.6073	**0.9662**	0.7170

Basing on macro-averaged plagdet, our algorithm shows the best results on 3 of 4 tests: copy&paste, paraphrased, and all types of plagiarism. However, the previous approach [Zubarev and Sochenkov, 2017] is still the best for the manual type. As for micro-averaged plagdet, our method turned out to be the best for paraphrased, manual, and all types of plagiarism. The baseline for copy&paste detection is very high, however, 3 of 4 models got very close to this result.

Models trained on the specific type of plagiarism show the best performance on their targets. Moreover, the model trained on the whole dataset not only achieved the best result on the whole test set (all types) but got very close to the best results on copy&paste and paraphrased parts. Also, it achieved the best micro-averaged plagdet for manual plagiarism.

However, for manual plagiarism the result turned out to be skewed (recall is high, but precision is low; or recall is low, but precision is high) for both current and previous approach, meaning that on this part of of the dataset even specific models fail to generalise appropriately. Additional work with detailed error analysis on manual plagiarism is required.

7. Conclusion and further directions

In this paper, we propose a new approach to Russian plagiarism text alignment and show how the diverse methods of Natural Language Processing can be successfully applied in one framework. The diversity of employed metrics perfectly matches heterogeneity of target problem. We use simple and standard metrics of textual similarity in combination with complex and modern ones (such as supervised sentence embeddings). This set allows to deal with simple and hard plagiarism cases at the same time.

A trained classifier unites the diverse metrics into a comprehensive structure. It allows teaching models focused on a specific type of plagiarism, keeping on the opportunity to train a universal one at the same time. This enlarges the possible set of problems to deal with and makes our approach flexible.

Finally, the algorithm shows great performance on all types of plagiarism in Russian language, significantly outperforming previous methods. This is the primary marker of its universality and high quality.

The main direction for improvement seems to be an enlargement and transformation of feature space to get better results on manual plagiarism detection. It could be achieved by taking into account syntactic similarity, adding word N-grams to token set similarities, and using more advanced approaches for classification, such as such CNN or LSTM models. Sentence embeddings feature set can be extended to include vectors from sent2vec [Pagliardini et al., 2017] and doc2vec [Le et al., 2014] models, which build distributed representations of sentences and short texts rather than words. Another serious improvement could be a reduction of hyperparameter space, such as classifier margin . Additional study on negative sampling in plagiarism detection may yield fruitful results here.

References

1. *Brlek A., Franjic P., and Uzelac N.* (2016), Plagiarism detection using word2vec model, Text Analysis and Retrieval 2016 Course Project Reports, pp. 4–7.
2. *Ferrero, J., Agnes, F., Besacier, L., & Schwab, D.* (2017), Using Word Embedding for Cross-Language Plagiarism Detection, arXiv preprint, arXiv, 1702.03082.
3. *Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T.* (2016), Fasttext.zip: Compressing text classification models, arXiv preprint, arXiv:1612.03651.
4. *Kutuzov, A., & Kuzmenko, E.* (2016), WebVectors: a toolkit for building web interfaces for vector semantic models, In International Conference on Analysis of Images, Social Networks and Texts, pp. 155–161.
5. *Larkham, P. J., & Manns, S.* (2002), Plagiarism and its treatment in higher education. Journal of Further and Higher Education, Vol. 26(4), pp. 339–349.
6. *Le H. T., Pham L. M., Nguyen D. D., Nguyen S. V., and Nguyen A. N.* (2016), Semantic text alignment based on topic modeling, In Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2016 IEEE RIVF International Conference on. IEEE, pp. 67–72.
7. *Maurer H., Kappe F., and Zaka B.* (2006), Plagiarism-a survey, J. UCS, Vol. 12(8), pp. 1050–1084.
8. *O'Brien, T. A., Kashinath, K., Cavanaugh, N. R., Collins, W. D., & O'Brien, J. P.* (2016), A fast and objective multidimensional kernel density estimation method: fastKDE, Computational Statistics & Data Analysis, Vol. 101, pp. 148–160.
9. *Potthast M., Stein B., Barron-Cede A., and Rosso P.* (2010), An evaluation framework for plagiarism detection, In Proceedings of the 23rd international conference on computational linguistics: Posters, Association for Computational Linguistics, pp. 997–1005.
10. *Resnik D. B.* (2005), The ethics of science: An introduction, Routledge.
11. *Sanchez-Perez M., Gelbukh A., Sidorov G., and Gomez-Adorno H.* (2017), Plagiarism detection with genetic-based parameter tuning. International Journal of Pattern Recognition and Artificial Intelligence, pp. 1860006.
12. *Smirnov I., Kuznetsova R., Kopotev M., Khazov A., Lyashevskaya O., Ivanova L., Kutuzov A.* (2017), Evaluation Tracks on Plagiarism Detection Algorithms for the Russian Language. International Conference "Dialogue 2017" Proceedings, Vol. 2, pp. 271–285.
13. *Sochenkov I. V., Zubarev D. V., Smirnov I. V.* (2017), The ParaPlag: Russian Dataset for Paraphrased Plagiarism Detection. International Conference "Dialogue 2017" Proceedings, Vol. 1, pp. 284–294.
14. *Sanchez-Perez M., Sidorov G., and Gelbukh A.* (2014), A winning approach to text alignment for text reuse detection at PAN 2014, In CLEF (Working Notes), pp. 1004–1011.
15. *Vani K., Gupta D.* (2017), Detection of idea plagiarism using syntax–semantic concept extractions with genetic algorithm, Expert Systems with Applications, Vol. 73, pp. 11–26.
16. *Wu, L., Fisch, A., Chopra, S., Adams, K., Bordes, A., & Weston, J.* (2017), StarSpace: Embed All The Things!, arXiv preprint, arXiv:1709.03856.

17. *Zubarev D. V., Sochenkov I. V.* (2017), Paraphrased Plagiarism Detection Using Sentence Similarity, International Conference “Dialogue 2017” Proceedings, Vol. 2, pp. 408–418.
18. *Pagliardini M., Gupta P., Jaggi M.* (2017), Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features, arXiv preprint, arXiv:1703.02507.
19. *Le Q., & Mikolov T.* (2014), Distributed representations of sentences and documents. In International Conference on Machine Learning (pp. 1188–1196).