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

Moscow, May 30—June 2, 2018

COMPARISON OF DEEP NEURAL NETWORK ARCHITECTURES FOR AUTHORSHIP ATTRIBUTION OF RUSSIAN SOCIAL MEDIA TEXTS

Dmitrin Y. V. (dmitrinyuri@gmail.com),
Botov D. S. (dmbotov@gmail.com),
Klenin J. D. (jklen@yandex.ru),
Nikolaev I. E. (Ivan_nikolaev@csu.ru)

Information Technologies Institute, Chelyabinsk State University, Chelyabinsk, Russian Federation

One of the important issues in modern computer forensics is authorship attribution i.e. identifying the author by inspecting various features of the document. It is a problem caused by a mostly anonymous format of communications over Internet, which allows for unhindered distribution of illegal or potentially dangerous information. To our knowledge there are no modern studies of this problem for documents in Russian. In this work we address the above-mentioned issue by applying a novel approach to automatic authorship attribution. The approach is based on deep neural networks, used to classify a document as written by a specific author. We also compare different methods of vector representations of the text such as label encoding and character n-grams. The experiment is carried out on several corpora in Russian, collected from several popular social media platforms. The presented experiment results compare several most popular neural network architectures, such as Recurrent Neural Network, Convolutional Neural Network, Long Short-Term Memory Network, and several hybrid models. The effectiveness of the proposed models is evaluated by considering the accuracy of classification.

Key words: deep learning, deep neural networks, authorship attribution, character n-grams, social media

СРАВНЕНИЕ АРХИТЕКТУР ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ В ЗАДАЧЕ ОПРЕДЕЛЕНИЯ АВТОРСТВА ТЕКСТОВ В СОЦИАЛЬНЫХ МЕДИА НА РУССКОМ ЯЗЫКЕ

Дмитрин Ю. В. (dmitrinyuri@gmail.com),
Ботов Д. С. (dmbotov@gmail.com),
Кленин Ю. Д. (jklen@yandex.ru),
Николаев И. Е. (Ivan_nikolaev@csu.ru)

Институт Информационных Технологий,
Челябинский Государственный Университет,
Челябинск, Российская федерация

Одной из важнейших проблем современной компьютерной криминалистики является определение авторства—установление автора путем анализа различных параметров документа. Это проблема вызвана преимущественно анонимностью коммуникации в Интернете, что позволяет беспрепятственно распространять незаконную или потенциально опасную информацию. Насколько нам известно, исследования данной проблемы с применением современных подходов к классификации текстов для документов на русском языке отсутствуют. В этой работе мы рассматриваем эту задачу, применяя подход к определению авторства на основе глубоких нейронных сетей. Мы также сравниваем различные методы векторных представлений текста, такие как кодирование меток и символьные n-граммы. Представлены результаты серии экспериментов, в которых сравниваются несколько наиболее популярных архитектур нейронных сетей, такие как рекуррентная нейронная сеть, сверточная нейронная сеть, LSTM и несколько гибридных моделей. Эксперимент проводится на нескольких корпусах на русском языке, собранных с популярных социальных сетей и блогов. Эффективность предлагаемых моделей оценивается с учетом точности классификации.

Ключевые слова: глубокое обучение, глубокие нейронные сети, определение авторства, символьные n-граммы, социальные медиа

1. Introduction

Author identification task for text documents published via Internet is one of the hottest issues in modern cybersecurity. User anonymity and irresponsible distribution of illegal materials regularly become subject of discussions at the highest level as being one of the most important issues of national security. Widespread use of social networks, instant messengers and blogs creates a great demand in the field of computer forensics for effective methods of authorship attribution for short texts.

Despite the recent development of text classification methods, based on deep neural networks, there are still almost no studies of their application to authorship identification of short Russian texts.

The main goal of this research is to analyze how various deep neural network architectures and text preprocessing approaches affect the accuracy of authorship identification for a variety of different corpora of short texts taken from Russian social media resources. Additionally we want to determine the effectiveness of the character n-gram models for Russian, which are being extensively studied for solving problem of text classification for other languages, mainly English. Another objective of the study is to determine how the accuracy of the proposed approaches to authorship identification depends on the size and number of instances of texts for each author.

2. Related Works

In recent years many researches have tackled the issue of authorship attribution and verification. Among their approaches, several major trends arose. First of these is a widespread usage of character-based features, caused, most likely by the types of document corpora, where authorship attribution is needed the most. Such corpora include a variety of short-text internet resources, like forum posts and comments, social network posts, and, most notable, twitter. Regardless of the overall method, corpus or language, majority of modern approaches seem to rely on character n-grams [1, 3–8, 10–13, 18, 20].

Character n-grams provide insight into various specifics of author's style, grammar and punctuation and overall topic of the text. In [3], Sapkota et al. present their analysis of importance of various types of character n-grams, grouping them both by their presumed function (morpho-syntax, thematic content, and style) and by their position in the document (affix, word, and punctuation). By training SVM (support vector machine) classifier on each of selected ten disjoint categories of n-grams, and comparing the resulting accuracy, authors conclude that prefix, suffix, space-prefix (prefix with a preceding space), and mid-word (n-gram from the middle of a longer word) are more important for a single domain task, while prefix, space-prefix, beg-punct (n-gram starting with punctuation mark), and mid-punct (n-gram containing punctuation mark between first and last character) show better results across domains. Among other features used to increase the quality of attribution are syntactic information [2, 18, 19], POS-tagging [5, 7, 17, 20], various frequency-based metrics [1, 2, 6–10, 12, 16, 17] etc.

Another notable trend, mostly present in works with comparatively large sets of features, is the usage of SVM classifiers. [1–5, 7–10, 12, 16] SVM seem to provide better or similar quality for the task of authorship attribution in comparison to other classic classification algorithms [2, 3, 4, 16].

However, with the recent growth in popularity of various neural network approaches, most machine learning approaches have since switched to applying those to many of the natural language tasks, including authorship attribution [11–13, 16, 18, 20].

Paragraph vectors trained on a combination of word n-grams, character n-grams, and POS-tag n-grams are used in [20]. Authors report, that their combined model surpasses algorithms based solely on word n-grams and character n-grams.

In [11] Sari et al. uses recently released continuous character embedding algorithm fastText [15] together with continuous vector embeddings for word n-grams, which are learned jointly with the classifier layer in a feed-forward NN. The authors demonstrate, that for the purposes of authorship attribution, character n-grams show better results in comparison to word n-grams and earlier, SVM-based approaches. Sari et al. uses four different corpora for their experiments: Judgment dataset consists of writing of three Australian High Court's judges on various topics with 902, 253 and 187 documents from each; CCAT-10—a corporate and industrial news of the Reuters Corpus volume 1 with 10 authors and 100 texts per author; CCAT-50 with 5,000 documents from 50 authors (only 50 texts used in training for each author); and IMDb62 dataset, which consists of 62,000 movie reviews and 17,550 message board posts from 62 users of the Internet Movie database.

Stamatatos [12] presents another character and word n-gram based approach, which attempts to apply distortion to the text—replacing a random subset of the most frequent words with a character mask, in order to remove topic-specific information from these documents. After masking a certain percentage of the words, the remaining document is used to extract word and character n-grams which are in turn used to train SVM classifier. Stamatatos uses 2 corpora—CCAT-10 and a Guardian corpus—articles and book reviews from The Guardian UK newspaper, with texts from 13 authors with topics of politics, society, UK, and world.

In [13] Solorio et al. suggest the use of convolutional neural networks to learn character representations and train to attribute text to authors. The authors also suggest a method for improving interpretability of achieved results, based on saliency scores, measuring sensitivity of the network to changes in each of the inputs, allowing researcher to create heatmaps of the input documents, showing segments, which were more important to the classifier decision, than others. For their experiments they use twitter corpus, containing about 9,000 twitter users with up to 1,000 tweets each. Limiting number of twits to 200, researchers vary number of authors from 100 to 1000.

Ferracane et al. [18] have applied several neural models, based on character-bigram CNNs, and investigated the impact of the discourse features on the quality of authorship attribution. Their results show that, while even the basic model achieves high performance, the addition of the secondary CNN for the extraction of discourse features improves the quality of the system, essentially making it new state-of-the-art, with the best model scoring 98.8 on F1.

Romanov and Meshcheryakov [16] utilize a variety of the frequency features to train a multilayer perceptron, a convolutional neural network and an SVM classifiers. They use messages from internet forum (forum.tomsk.ru) as a corpus for experiments. It has a total of 10 authors with about 144 texts per author. Authors specify, that they only classify each text in a binary fashion, rather than a multiclass.

3. Approach

Our approach consists of two stages: document representation and the machine learning process itself. Below we describe both of these components in detail. Code implementation of our experimental setups is available on GitHub[1].

3.1. Word representation

There exists a variety of different approaches to numeric vector representation of text documents. Below we take a look at basic methods used to supply inputs of neural networks, classic frequency-based and topic-modelling techniques, and modern embedding approaches.

Label Encoding is a method of text-hashing, which assigns a unique identifier that serves as a label to each categorical variable. The main disadvantage of this approach is that some algorithms of machine learning might consider one feature more significant than others.

One Hot Encoding is the approach to of text-hashing, in which categorical variables are converted to a binary vector. Generally categorical variables are represented in the form of a numeric zero-filled vector, where only the integer index, corresponding to a specific category, is assigned a one.

Under this approach, each text can then be considered a disjunction of such vectors for each of the words used. However, this results in extremely sparse binary vectors, while also losing any additional information, such as the number of occurrences of each word or their order.

Partially this problem can be dealt with by replacing ones with TF-IDF weights of each word, which would add frequency information to one-hot vectors, however that will not affect the sparseness which could significantly increase the training time of the neural network.

Another approach allows to mitigate both the lack of structural information, and the sparseness of the one-hot vectors. N-grams break text down into continuous groups of n tokens: characters or words. Since the number of character n-grams is limited for any language (especially for smaller n) the vector representations become much denser, having less dimensions.

Some attempts have been made to achieve reduced vector space sparseness. A selection of topic modelling techniques can be used to represent documents as vectors in the space, which dimensions are various topics, covered by the document collection. Latent Semantic Indexing utilizes singular value decomposition on term-document matrix, while Latent Dirichlet Allocation and Probabilistic LSI attempt to treat document collection and each text in it as a probabilistic distribution of a number of topics.

More recently a variety of modern methods have been suggested, which attempt to reduce the dimensionality of vector representations, while incorporating additional information. These methods are generally referred to by the umbrella term “word embedding”, reflecting the mathematical meaning of embedding one vector space into another.

[1] https://github.com/XATTABrus/Authorship-Attribution

One of the methods, word2vec [14] has quickly gained popularity among researchers in the field. Word2Vec uses a shallow neural network, which learns to connect words with their context, resulting in a set of numeric vectors of desired dimensionality. By learning the connection between words and context words, the resulting vectors tend to be closer for words, appearing in similar contexts and further apart for words that do not share many contexts.

3.2. Deep Neural Network Architectures

Here we provide a set of brief descriptions of the deep neural network architectures, typically used for text classification problems. We later use these in our experiments.

Recurrent Neural Network (RNN) are networks containing feedbacks and allowing information to be stored. However, these networks have a significant disadvantage, known as a vanishing gradient, which causes them to be unable of storing information for a long time.

Long short-term memory (LSTM) is a special kind of architecture for recurrent neural networks, capable of learning long-term dependencies, unlike simple recurrent neural networks.

Bidirectional LSTM (BiLSTM) is a modification of the LSTM network, in which a sequence from the beginning and the end is sent to the network at the same time.

Convolutional Neural Network (CNN) is a kind of deep neural network based on the operations of convolution and pooling. This type of neural network is widely used in the classification of images, audio signals, as well as texts.

In addition to the standard deep neural network architectures that have been described above, various combinations of architectures are often used, for example, the combination of several LSTM layers in succession, or CNN with a gradual decrease in the number of filters, in an attempt to isolate more general patterns. Typical solution to the classification problem is a fully connected layer at the end of the network, which determines the class value for the inputs.

3.3. Model Parameters

The first layer of each network is embedding layer with different parameters of embeding_size (depending on the corpus). The output size of embedding-layer is 200, the next layer after embedding, is spatial dropout with parameter 0.2. The last layer is fully connected with the softmax activation function. Loss function is categorical cross-entropy, optimizer is adam with default values, and the metric is accuracy. The average number of learning epochs is 10.

After optimizing parameters for our networks, we settled on the following:

- Dense:
  - Number of neurons 200
  - Dropout 0.2
- CNN:
  - Number of convolution filters 512

  - Kernel size 3
  - Activation function RELU
  - GlobalMaxPooling
- CNN + CNN:
  - Number of convolution filters 512
  - Number of convolution filters 256
  - Kernel size 3
  - Activation function RELU
  - GlobalMaxPooling
- LSTM:
  - 128 hidden layers
  - Dropout 0.3
  - Recurrent_dropout 0.3
- Bidirectional LSTM:
  - 128 hidden layers
  - Dropout 0.3
  - Recurrent_dropout 0.3
- LSTM + CNN и CNN + LSTM
  - Number of convolution filters 512
  - Kernel size 3
  - Activation function RELU
  - GlobalMaxPooling
  - 128 hidden layers
  - Dropout 0.3
  - Recurrent_dropout 0.3

It should also be noted that when using a label encoding, the documents are separated into tokens by whitespace, the tokens are then converted to the lower case, and punctuation is removed. With character n-grams, the text remains in its original state.

4. Datasets

For our experiments, we make use of 3 datasets (table 1):

- Habrahabr blog[2], which contains texts of an average length of 2,000 words. These texts are mostly IT-related and thus have a domain-specific vocabulary.
- Vk.com (VK)[3], which contains posts from vk.com social network, with an average length of 100 words. These have a chaotic nature and an open-domain vocabulary.
- Echo.msk.ru (Echo)[4], which contains various blog posts from mediaportal Echo Moscow. The publications are on various topics with the average length of 2000 words.

---

[2] https://goo.gl/naUXcv

[3] https://goo.gl/zR7rwB

[4] https://goo.gl/zzhH7p

There are currently no suitable datasets publicly available for the Russian. The closest short text corpus is the collection of microblog messages from Twitter, but this dataset suffers shortcomings due to messages being extremely short, only up to 140 characters and a vast amount of non-filterable advertising materials.

Because of this, it was decided to collect our own corpora. As examples of large documents, we chose articles from Habrahabr.ru, while posts from VK.com were chosen as set of short texts.

Table 1: Corpora parameters

Corpus	# of documents	# of authors	# of tokens	# of unique tokens
Habrahabr	3,400	30	4,500,000	270,000
VK 50	6,000	50	600,000	94,000
VK 100	9,000	100	910,000	126,000
Echo 50	28,000	50	14,000,000	1,020,000
Echo 100	42,000	100	22,000,000	1,400,000

For the experiment, each corpus was broken down into the training and test set with the 80% to 20% ratio. To assess the quality we applied accuracy metric.

Habrahabr dataset is shown in more detail in figure 1, which shows the number of words per document, and figure 2, which shows the number of publications per author.

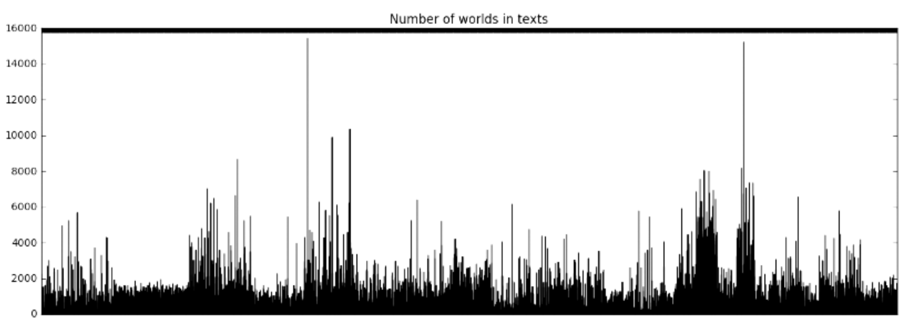

Fig. 1: Words per document in Habrahabr corpus

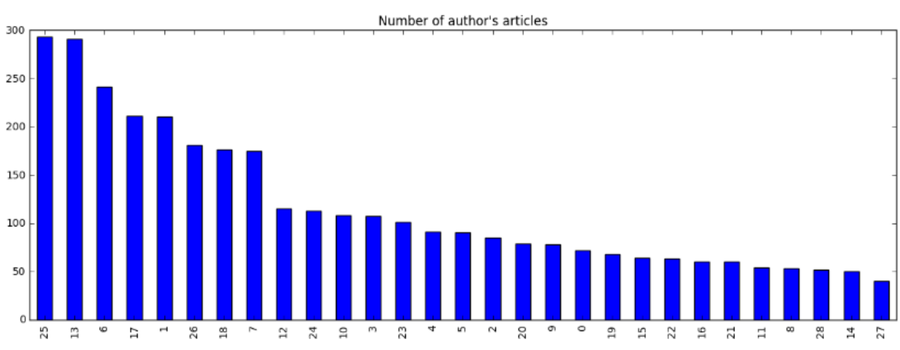

Fig. 2: Documents per author in Habrahabr corpus

To test the stability of selected methods to increasing number of classes, we divided VK and Echo datasets into 2 parts. The first part consists of 50 authors, the second part contains 100 authors. Figures 3–6 show more details for these subsets. The same has been done to Echo dataset, which is shown in greater detail in figures 7–10.

The VK corpus contains messages posted by users on their personal pages, excluding reposts, spam messages, and any messages containing quotes, jokes, and advertisement.

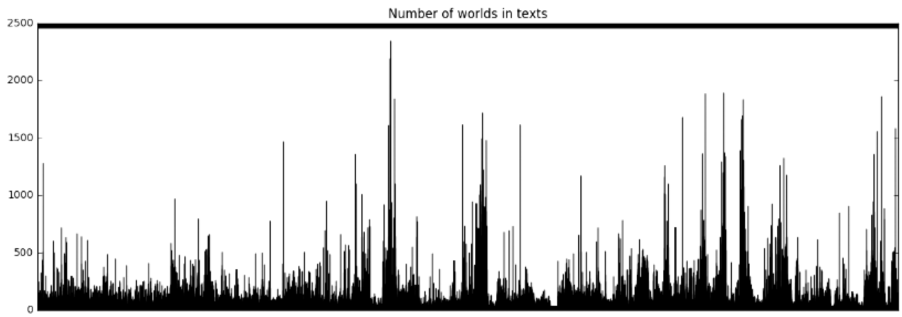

Fig. 3: Words per document in VK50 corpus (with 50 authors)

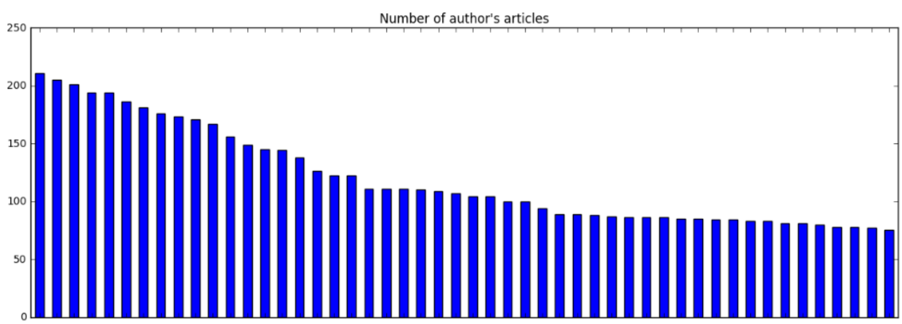

Fig. 4: Documents per author in VK50 corpus (with 50 authors)

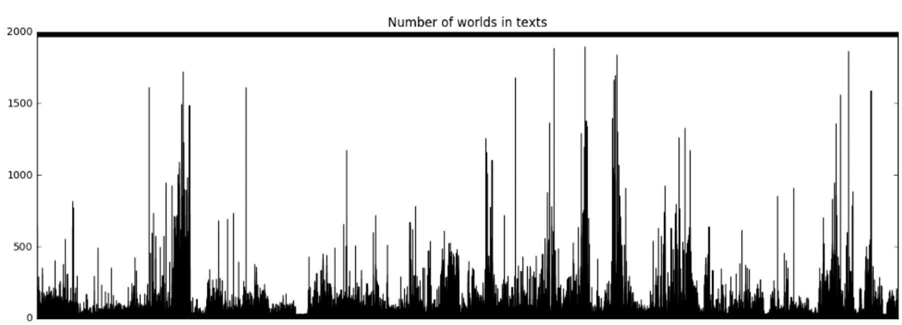

Fig. 5: Words per document in VK100 corpus (with 100 authors)

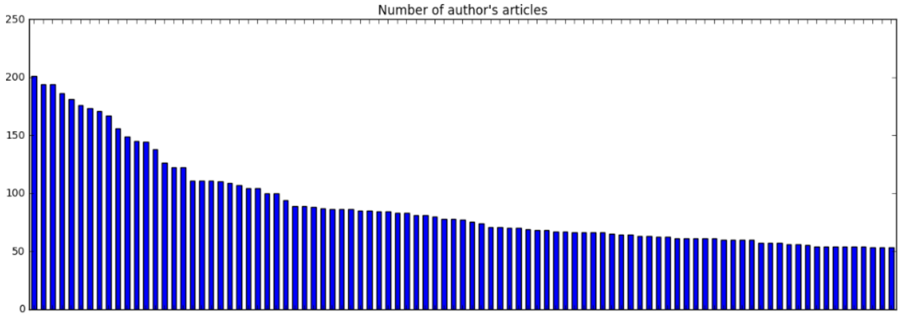

Fig. 6: Documents per author in VK100 corpus (with 100 authors)

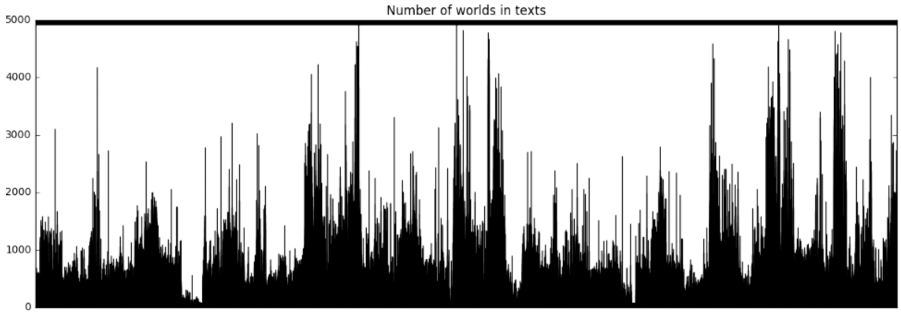

Fig. 7: Words per document in Echo50 corpus (with 50 authors)

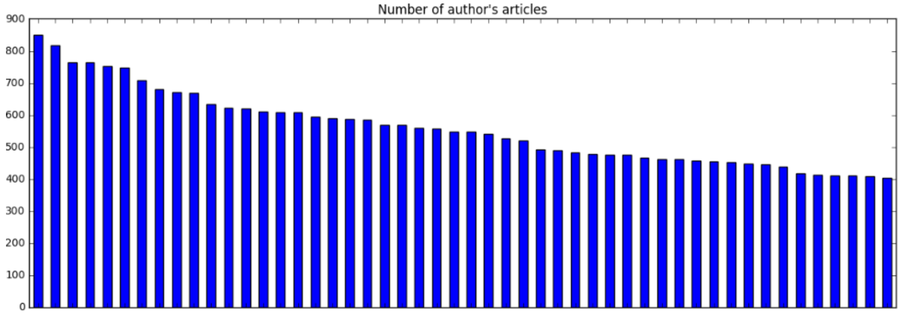

Fig. 8: Documents per author in Echo50 corpus (with 50 authors)

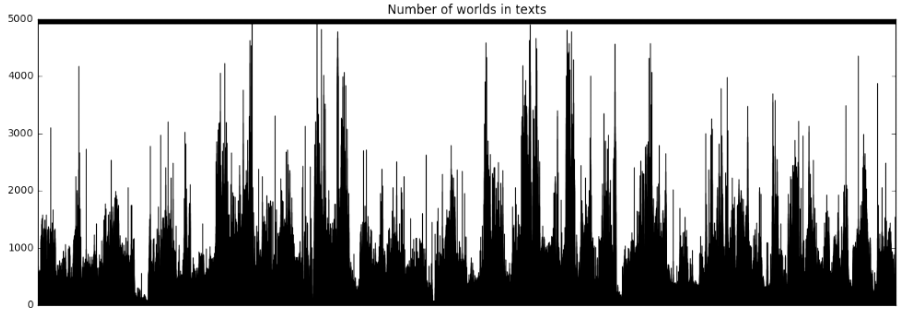

Fig. 9: Words per document in Echo100 corpus (with 100 authors)

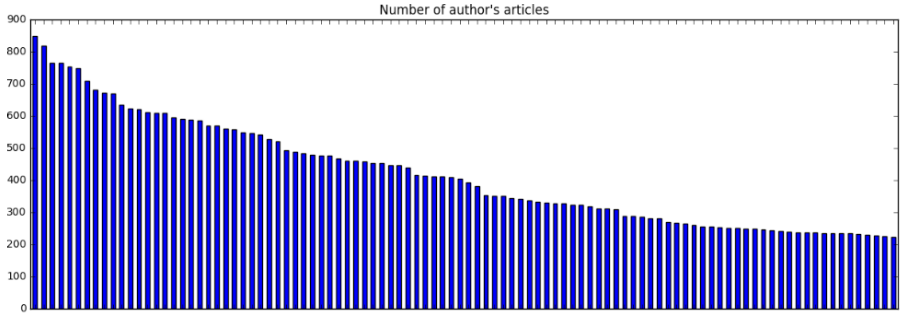

Fig. 10: Documents per author in Echo100 corpus (with 100 authors)

It is also worth noting that any text preprocessing, that involves removing some of the characters, will ultimately reduce the quality of the classification in this approach, as it will reduce the amount of available data on the personal style, used by an author.

5. Experiments and Evaluation

As a baseline for the task of authorship attribution, the classical neural network architecture was applied—a fully-connected feedforward network. Table 2 shows the baseline results after parameter tuning for VK100 corpus, using character 3-grams. The choice was made according to the following parameters: the number of neurons, the number of hidden layers, the use of Dropout regularization

Table 2: Baseline accuracy on VK100 corpus

Dataset	Dense (200)	Dense (500)	2x Dense (200)	2x Dense (200) Dropout	3x Dense (200) Dropout
VK 100	0.3165	0.3132	0.2799	0.3065	0.2866

It should be noted that because of the high variance of the word counts among different documents, we make use of pad_sequences (dropping of long texts and zero-padding of short texts) to mitigate this.

For each corpus a variety of neural network architectures and text representation algorithms were used.

First we compare accuracy of various deep neural network architectures, for the task of authorship attribution on Habrahabr and VK datasets. These experiments also show the difference in quality caused by using either label encoding or character 3-gram representation as an input to each of the algorithms.

Table 3: Accuracy in experiments on Habrahabr corpus

Vectorizer	LSTM	Bi LSTM	CNN	CNN + CNN	CNN + LSTM	LSTM + CNN
Label Enc.	—	—	0.8139	0.6470	0.5022	0.5072
3-Grams	—		**0.8744**	0.8331	0.7790	0.7839

Table 4: Accuracy in experiments on VK50 corpus

Vectorizer	LSTM	Bi LSTM	CNN	CNN + CNN	CNN + LSTM	LSTM + CNN
Label Enc.	0.4431	0.4323	0.5103	0.4895	0.4586	0.4038
3-Grams	0.3514	0.2947	**0.5868**	0.5326	0.3957	0.4699

Table 5: Accuracy in experiments on VK100 corpus

Vectorizer	LSTM	Bi LSTM	CNN	CNN + CNN	CNN + LSTM	LSTM + CNN
Label Enc.	0.3509	0.3404	0.4446	0.3819	0.3520	0.3004
3-Grams	0.3043	0.2517	**0.5255**	0.4667	0.3093	0.3614

Values given in Tables 3–5 show, that the following deep neural network architectures provide on average similar level of accuracy—LSTM, Bi LSTM, CNN + LSTM, LSTM + CNN. The best results for blogs and social network posts are achieved by CNN.

Next, in order to further evaluate the convolutional neural network, we experiment with different vector representations as an input for CNN (Table 6).

Table 6: Accuracy of CNN on various vector representations

Encoding	VK100	Habrahabr	Echo50	Echo100
Lable Enc.	0.4446	0.8139	0.8013	0.7355
3-Grams	**0.5255**	0.8744	0.8622	**0.8359**
4-Grams	0.4928	**0.8789**	**0.8853**	0.8217

Results, presented in Table 6, show the effectiveness of character n-grams over label encoding. 3-gram representations show themselves significantly better for short texts from social networking sites, while for longer blogs both 3-gram and 4-gram representations achieve similar accuracy.

We further investigate the effectiveness of training CNN with character 3-gram representations using varying number of documents per author. We also compare these values depending on whether or not we keep punctuation while applying character 3-grams to the text.

Table 7: Accuracy of CNN depending on the number of examples per author for VK100 dataset

# of documents	10	20	30	40	50
Accuracy w/o punctuation	0.2900	0.3425	0.4150	0.4012	0.4570
Accuracy w. punctuation	0.2150	0.3550	0.3683	0.3837	0.4110

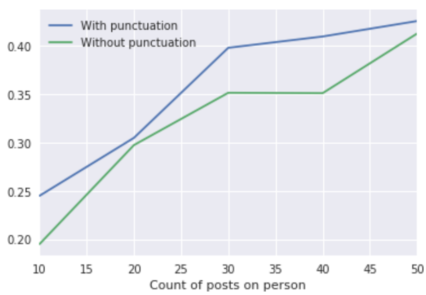

Fig. 11: Accuracy of CNN depending on the number of examples per author for VK100 dataset

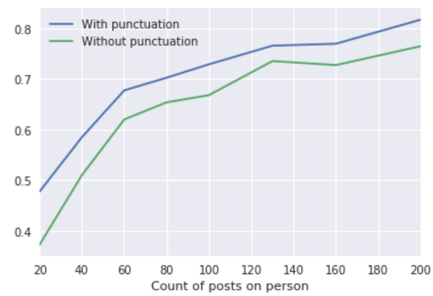

Fig. 12: Accuracy of CNN depending on the number of examples per author for Echo100 dataset

Table 8: Accuracy of CNN depending on the number of examples per author for Echo100 dataset

# of documents	20	40	60	80	100	130	160	200
Accuracyw/o punctuation	0.3725	0.5100	0.6200	0.6538	0.6680	0.7354	0.7275	0.7648
Accuracy w. punctuation	0.4775	0.5850	0.6775	0.7025	0.7290	0.7658	0.7697	0.8173

In Figures 11 and 12, we show how the accuracy changes with increasing number of examples per author. At the same time, keeping punctuation marks in the character n-gram representation improves accuracy by an average of 5–6%.

The results of analysis of different convolution kernel sizes are presented in Table 9. The best quality is achieved by a convolutional neural network with a kernel size of 3.

Table 9: CNN kernel size with character n-gram encoding

# of cnn kernel	2	3	4	5	10
VK100	0.5227	**0.5327**	0.5166	0.5166	0.4895

6. Conclusion

In this paper we present an analysis of various deep neural network architectures, vector representations of the text, for the task of authorship attribution. The study was conducted on Russian corpora of various social media such as blogs and social network posts.

When choosing optimal parameters for neural networks, CNN has proved to show the best results for experimental data, while character n-grams with the inclusion of punctuation marks has turned out to be the best way of texts representation. We assume that the author's style cannot be analyzed as just a sequence of words as it is in the case of the LSTM, but should be considered a more complex entity that the convolutional neural network can best identify. This is confirmed both by our experiments with Russian, and also by the experiments with English in [13]. Sequence analysis shows good results in tasks of text classification by meaning, machine translation or conversational systems. These results are consistent both in relation to short social media posts (averaging 100 words per post) and longer texts of blog posts (an average of 2000 words per document).

In the future, in order to further improve the quality of authorship attribution, we propose using a hybrid classification model utilizing both the convolutional neural networks over character n-grams and a separate classifier over the linguistic features of each text.

References

1. *H. J. Escalante, T. Solorio, and M. Montes-y-Gómez.* Local histograms of character n-grams for authorship attribution. In Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 288–298, 2011.
2. *G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona Hernandez.* Syntactic n-grams as machine learning features for natural language processing. Expert Syst. Appl., 41(3):853–860, 2014.
3. *U. Sapkota, S. Bethard, M. Montes-y Gomez, and T. Solorio.* Not all character n-grams are created equal: A study in authorship attribution. In Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 93–102, 2015.
4. *M. Popescu and C. Grozea.* Kernel methods and string kernels for authorship analysis. CLEF 2012 Evaluation Labs, 2012.
5. *A. Bartoli, A. Dagri, A. D. Lorenzo, E. Medvet, and F. Tarlao.* An author verification approach based on differential features. CLEF 2015 Evaluation Labs, 2015.
6. *R. Layton, S. McCombie, and P. Watters.* Authorship attribution of IRC messages using inverse author frequency. In Cybercrime and Trustworthy Computing Workshop, pages 7–13, 2012.
7. *T. Qian, B. Liu, L. Chen, and Z. Peng.* Tri-training for authorship attribution with limited training data. In Annual Meeting of the Association for Computational Linguistics, pages 345–351, 2014.

8. *S. Afroz, A. Caliskan-Islam, A. Stolerman, R. Greenstadt, and D. McCoy.* Doppelganger finder: Taking stylometry to the underground. In IEEE Security and Privacy, 2014.
9. *R. M. Green and J. W. Sheppard.* Comparing frequency-and style-based features for Twitter author identification. In FLAIRS Conference, 2013.
10. *R. Schwartz, O. Tsur, A. Rappoport, and M. Koppel.* Authorship attribution of micro-messages. In Conference on Empirical Methods on Natural Language Processing, pages 1880–1891. ACL, 2013.
11. *Y. Sari, A. Vlachos, and R. M. Stevenson,* (2017) Continuous N-gram Representations for Authorship Attribution. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. European Chapter of the Association for Computational Linguistics (EACL 2017), 3–7 April 2017.
12. *E. Stamatatos.* Authorship Attribution Using Text Distortion. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 1138–1149, Valencia, Spain, April 3–7, 2017.
13. *T. Solorio, P. Rosso, M. Montes-y-Gómez, P. Shrestha, S. Sierra, F. A. González.* Convolutional Neural Networks for Authorship Attribution of Short Texts. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 669–674, Valencia, Spain, April 3–7, 2017.
14. *T. Mikolov, K. Chen, G. Corrado, and J. Dean.* Efficient estimation of word representations in vector space. International Conference on Learning Representations (ICLR), Workshop.
15. *A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov.* Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759, 2016.
16. *A. Romanov, R. Meshcheryakov.* Identification of authorship of short texts with machine learning techniques. In Proceedings of the Dialog'2010. pp. 407–413, 2010.
17. *T. Litvinova, O. Litvinlova, O. Zagorovskaya, P. Seredin, A. Sboev, and O. Romanchenko.* "Ruspersonality": A Russian corpus for authorship profiling and deception detection. International FRUCT Conference on Intelligence, Social Media and Web (ISMW FRUCT), pp. 1–7, 2016.
18. *E. Ferracane, S. Wang, and R. J. Mooney.* Leveraging Discourse Information Effectively for Authorship Attribution. Proceedings of the The 8th International Joint Conference on Natural Language Processing, pp. 584–593, 2017.
19. *J. Bjorklund, N. Zechner.* Syntactic methods for topic-independent authorship attribution. Article in Natural Language Engineering 23(05), pp. 789–806, September 2017.
20. *H. Gómez-Adorno, J-P. Posadas-Durán, G. Sidorov, D. Pinto.* Document embeddings learned on various types of n-grams for cross-topic authorship attribution. Journal "Computing", pp. 1–16, 2018.