Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 123 (2018) 424–431

Procedia Computer Science

www.elsevier.com/locate/procedia

8th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2017

Deep Learning neural nets versus traditional machine learning in gender identification of authors of RusProfiling texts

Alexander Sboev[1,2], Ivan Moloshnikov[1], Dmitry Gudovskikh[1], Anton Selivanov[1], Roman Rybka[1], and Tatiana Litvinova[1,3]

[1] National Research Centre 'Kurchatov Institute', Moscow, Russian Federation
[2] MEPhI National Research Nuclear University, Moscow, Russian Federation
[3] Voronezh State Pedagogical University, Voronezh, Russian Federation
sag111@mail.ru, ivan-rus@yandex.ru, dvgudovskikh@gmail.com, aaselivanov.10.03@gmail.com, rybkarb@gmail.com, centr_rus_yaz@mail.ru

Abstract

In this paper we compare accuracies of solving the task of gender identification of RusProfiling texts without gender deception on base of two types of data-driven modeling approaches: on the one hand, well-known conventional machine learning algorithms, such as Support Vector machine, Gradient Boosting; and, on the other hand, the set of Deep Learning neuronets, such as neuronet topologies with convolution, fully-connected, and Long Short-Term Memory layers, etc. The dependence of effectiveness of these models on the feature selection and on their representation is investigated. The obtained F1-score of 88% establishes the state of the art in the gender identification task with the RusProfiling corpus.

Peer-review under responsibility of the scientific committee of the 8th Annual International Conference on Biologically Inspired Cognitive Architectures

Keywords: gender identification, neural networks, natural language processing, data-driven modeling

1 Introduction

In the current practice there are two classes of data-driven models applicable to the gender identification task: conventional machine learning algorithms like Support Vector Machine (SVM) ,Gradient boosting and the set of Deep Learning neuronets with the convolution, the long shortterm memory layers (LSTM), etc. The contemporary levels of accuracies, reached on the task are different for different languages. Results of the PAN 2017 competition [1] give the main milestones in the question what is the state of the art for such languages as Arabic, English,Portuguese, Spanish. One of the PAN 2017 tasks was the gender identification of Twitter texts. The competition corpus consisted of 500 tweets of 100 authors. The dataset was divided into 60% (300 texts) for training and 40% (200 texts) for testing. The best results for English and Spanishwere obtained using the Linear Support Vector Machine, see [2]. Input texts were

Peer-review under responsibility of the scientific committee of the 8th Annual International Conference on Biologically Inspired Cognitive Architectures
10.1016/j.procs.2018.01.065

represented as frequency vectors of character 2,3,4-grams and word forms, such as bigrams and trigrams.Then the frequencies were replaced by the measure $1 + log(term - frequency)$, which showed the best result. For the Arabic language, the best model [3] was based on a representation of text as a vector containing combinations of character, word and POS n-grams with emojis, character flooding, sentiment words. Logistic regression was used for training the classifier. Deep learning approach in [4] showed the best result on the Portuguese language. The authors applied Recurrent Neural Networks (RNN) for words and Convolutional Neural Networks (CNN) for characters. Thereby, two representations of different levels for a single message were obtained. They were combined and classified by gender, using attention mechanism, max-pooling layer, and fully-connected layer. The word embeddings layer was preliminarily trained with the skip-gram. For character embedding layer, weights were randomly initialized with a uniform distribution.

So, according to the results of the competition [3], traditional methods of ML showed the best result for English, Spanish, and Arabic languages, while deep learning was the winner for Portuguese. The aim of this paper is to find out which approach would be better for Russian. We use different corpora, described in details in section 2, with different sets of input features described in section 3.1. Among the corpora created with offline respondents, there was one collected by crowdsourcing on the Web, and another goal of the paper was to compare the results obtained on different corpora and to assess the possibility of further use of crowdsourcing. The algorithms we use, traditional machine learning ones and deep learning topologies, are presented in section 3.2. Results of calculations and their discussions are in sections 4 and 5.

Part name	**Total**	**Men**	**Women**	**Avg. length(M)**	**Avg. length(W)**	**Male authors**	**Female authors**
GI(or)	394	125	269	123.38	107.82	47	92
GI(bal)	250	125	125	123.38	111.19	47	74

Table 1: Statistics of the Gender imitation text corpus.

Part name	**Total**	**Men**	**Women**	**Avg. length(M)**	**Avg. length(W)**	**Male authors**	**Female authors**
GI_cs(or)	3204	1161	2043	75.16	77.21	280	438
GI_cs(bal)	2322	1161	1161	75.16	78.89	280	265
GI_cs_A_B(or)	2134	772	1362	75.35	77.21	279	438
GI_cs_A_B(bal)	1544	772	772	75.35	78.91	279	264

Table 2: Statistic of the Gender imitation crowdsource text corpus.

Part name	**Total**	**Men**	**Women**	**Avg. length(M)**	**Avg. length(W)**	**Male authors**	**Female authors**
original	1033	641	392	60.42	66.49	641	392
balanced	784	392	392	58.61	66.49	392	392

Table 3: Statistics of the RusPer text corpus.

Part name	Total	Men	Women	Avg. length(M)	Avg. length(W)	Male authors	Female authors
Reviews(or)	1033	641	392	60.42	66.49	641	392
Reviews(bal)	784	392	392	58.61	66.49	392	392
FB(or)	250	136	114	1381.87	1395.82	136	114
FB(bal)	228	114	114	1382.58	1395.82	114	114
FB_split(or)	1617	868	749	197.36	195.19		
FB_split(bal)	1498	749	749	200.86	195.19		
Tw(or)	1541	998	543	1721.91	1638.18	998	543
Tw(bal)	1086	543	543	1746.76	1638.18	543	543
Tw_split(or)	7512	5062	2450	298.37	315.13	-	-
Tw_split(bal)	4900	2450	2450	298.45	315.13	-	-
LJ(or)	11	6	5	52812.5	63049.8	6	5
LJ(bal)	10	5	5	50463	63049.8	5	5
LJ_split(or)	3256	1632	1624	193.91	193.74	-	-
LJ_split(bal)	3248	1624	1624	164.27	163.74	-	-

Table 4: Statistics of the RusProfiling text corpus.

2 Used corpora

We used several corpora, having different numbers of authors, texts, and text length. For this reason the original corpora were balanced by the number of texts by women and men. The statistics of the used datasets, original and balanced, is presented in Tables 1-4. The Russian **Gender imitation (GI)** (Table 1) corpus consists of specially written author texts[5]. Each author chose a topic from a list of four: a letter to a friend, a self-description for a dating site, a complaint about the boss or about a tour, and then was instructed to write three texts on the same topic: text A - in the author's natural style, text B - as someone of the opposite sex, text C - as someone else of the same sex. **GI crowdsource (GI_cs)** (Table 2) contains texts collected by crowdsourcing platform to extend the GI corpus. It was collected the same way and has the same subsets as GI. We singled out the subcorpus of texts: **GI_cs_a_b** denotes the part of GI_cs without the C texts (similar to GI type C). **RusPer (RusPer)**(Table 3) is a Russian-language corpus [6] of written texts labeled with data on their authors. Each text has a metadata like gender, age, personalities, education level, neuropsychological testing data, etc. This paper uses a part of the corpus, consisting of texts on two topics: letters to a friend and picture descriptions. **RusProfiling (RusProf)**(Table 4) corpus [7] contains texts collected from different social media platforms as Twitter, Facebook and LiveJournal, along with **Reviews** - the set of reviews collected manually. This data set was divided into subsamples, which allows to study how the features of the corpus influence the quality of the models and to evaluate the possibilities of cross-genre classification. Some subsamples of RusProf were divided into two sets. In the first set all messages were merged into one text for each author: **Twitter(Tw)** - 1000 twitter messages for each user, **LJ** - the set of blog texts from LiveJournal, **FB** - messages from FaceBook walls. In the second set, after combining each user messages into one text, it was separated into documents of 15 sentences. These subsamples are denoted as **Tw_split**, **LJ_split**, and **FB_split**. We also conducted an evaluation of the models with the corpora preliminarily balanced by the author's gender. These results are noted 'original (or)' or 'balanced(bal)' respectively next to the name of the corpus in the tables further.

3 Methods

3.1 Features

The following sets of features were used in our work in combination with different models:

TF-IDF for n-grams(TF-IDF): The vector of n-gram frequencies, characterizing a document d, is obtained by applying by the TF-IDF formula $\mathrm{tf}-\mathrm{idf}(t,d) = \mathrm{tf}(t,d) \times \mathrm{idf}(t)$ to each n-gram t of the collection D. Here t is a character n-gram from 3 to 8 characters long, $\mathrm{tf}(t,d)$ is the number of times t occurs in d; $\mathrm{idf}(t) = \log \frac{1+n_d}{1+\mathrm{df}(d,t)} + 1$. Here n_d is the total number of texts, and $\mathrm{df}(d,t)$ is the number of texts that contain t. Only n-grams that exist in more than 100 texts are used. We use different values of minimal $\mathrm{df}(d,t)$ in the range from 1 to 10.

Char n-grams(Char). To represent a document as a vector, the frequency of encountering symbolic n-grams ranging from 1 to 8 is calculated.

GRM-1: each word is encoded by a 49-dimensional binary vector of morphological properties. The size of a document is fixed at 300 words: longer texts are clipped, and shorter texts are extended with null vectors.

GRM-2: each word is again encoded by a 49-dimensional morphological properties vector, but then all the vectors of a text are concatenated, forming a vector of the length 14700.

Word2vec. This is an already existing model that was preliminarily trained on a collection of random Russian web pages crawled in December 2014, contains 9 million documents in total. Corpus size is $660,628,738$ tokens. Model was trained using the Continuous Skip-Gram algorithm. Vector dimensionality was set to 500, window size 2. Lemmas occurring less than 30 times were ignored [8].

Sequences feature (Seq. feat): Texts are represented as sequences of words with a full set of morphological tags (person, gender, part of speech, etc.) along with the type of syntactic relation with a parent, all one-hot encoded.

3.2 Models

3.2.1 Conventional machine learning methods

Support vector machine (SVM). We used the classifier based on a support vector machine with linear kernel. The following hyperparameters were used: regularization parameter $C = 1$, $L2-norm$ used in the penalization and squared hinge-loss function. **Gradient boosting(GB)** classifier were trained with the following parameters: learning rate is 0.05, the number of boosting stages to perform is 300, the minimum number of samples to split an internal node is 19, the maximum depth of the tree is 12.

3.2.2 Neural networks topology

Model 1 is based on convolution and fully connected layers: **Convolutional Neural Network (CNN)**: 128 neurons, window size = 2, activation function = Relu;**Maxpooling layer**: window = 4, step = 4; **CNN**: 128 neurons, window = 2, activation function = Relu; **Maxpooling layer**: window = 4, step = 4; **CNN**: 128 neurons, window = 2, activation function = Relu; **Maxpooling layer**: window = 4, step = 4; **CNN**: 128

neurons, window = 2, activation function = Relu; **GlobalMaxPooling**; **Dropout 0.5**; **Fully-connected MLP layer**: 128 neurons, activation function = Relu; **Dropout 0.5**; **Output layer - MLP**: activation function = softmax.

Model 2 . The concept of Stacked Bidirectional LSTM neural network architecture was inspired by [9], who showed the possibility of using such an architecture for character prediction, machine translation and image classification tasks. The peculiarity of this network worth noting is that the first branching on LSTM returns sequences that are further concatenated. It consists of the following layers: **CNN**: 30 neurons, window size = 3, activation function = ReLU. The input is padded so that the output has the same length as the original input (futher denoted as padding = 'same'); **Maxpooling layer**: window = 2; **CNN**: 30 neurons, window size = 3, activation function = Relu, padding = 'same'; **Maxpooling layer**: window = 2; **CNN**: 30 neurons, window size = 3, activation function = Relu, padding = 'same'; **Maxpooling layer**: window = 2; **leftLSTM**: 30 neurons, and the last output in the full sequence is returned (further denoted by 'return sequence') and in parallel **rightLSTM**: 30 neurons, the input sequence is processed backwards and the reversed full sequence is returned (further denoted as 'go backwards'); **Concatenate**: leftLSTM and rightLSTM; **leftLSTM_2**: 30 neurons and in parallel **rightLSTM_2**: 30 neurons, "go backwards"; **Concatenate**: leftLSTM_2 and rightLSTM_2; **Dropout 0.5**; **Dense**: 10 neurons, activation function = "tanh"; **Dropout 0.5**; **Output layer - Dense**: 2 neurons, activation function = 'softmax'.

Model 3 . The architecture of this neural network has been adapted from the work [10]. Network topology: **CNN**: 30 neurons, window size = 3, activation function = Relu. The input is padded so that the output has the same length as the original input (futher denoted as padding = "same"); **Maxpooling layer**: window = 2; **CNN**: 30 neurons, window size = 3, activation function = Relu, padding = "same'; **Maxpooling layer**: window = 2; **CNN**: 30 neurons, window size = 3, activation function = Relu, padding = 'same';**Maxpooling layer**: window = 2; **leftLSTM**: 30 neurons and in parallel **rightLSTM**: 30 neurons, "go backwards"; **Concatenate**: leftLSTM and rightLSTM; **Dropout 0.5**, **Dense**: 5 neurons, activation function = 'tanh'; **Dropout 0.5**; **Output layer - Dense**: 2 neurons, activation function = 'softmax'.

Network training was performed with early stopping (if the error on the validation set grows during 15 epochs, then stop). After training the weights from the best model were loaded. We used the Rmsprop algorithm [11] and learning rate of 0.001.

4 Experiments

Evaluations of F1 score (F1_mean) and mean deviations (F1_std) were obtained. In case of Conventional machine learning alghoritms all the training data have been used to fit models. F1_std was calculated on base of 10 different cycles of training and testing of single configuration. The training dataset was split 5 times into 80% to train and 20% to validate. The F1_mean and the F1_std were calculated on the base of these split data sets. The baseline was calculated according to the size of classes, and was F1-score=0.59 for the original, unbalanced test dataset and F1-score=0.5 for the balanced test dataset. Results obtained with training on 'a' and 'b' GI subsets are of especial interest, in spite of their low accuracy. By excluding the 'c' subset we balance the training set by the number of actual-gender samples and opposite-gender-mimicking samples. The reason is that, when actual-gender samples ('a' and

#	Train datasets	Model	Features	F1	F1_std	Delta *(bl 0.59)*
1	RusPer; GI A,B,C	Model 2	GRM-1	0.73	0.04	0.14
2	RusPer; GI A,B,C	Model 3	GRM-1	0.76	0.07	0.17
3	RusPer; GI A,B	Model 2	GRM-1	0.65	0.1	0.06
4	GI_cs; GI A,B,C	SVM	Char	0.74	0.04	0.15
5	GI_cs; GI A	GB	GRM-2	0.76	0.19	0.17
6	RusPer	Model 3	GRM-1	0.78	0.04	0.19
7	GI_cs_a	GB	TF-IDF	0.74	0.03	0.15
16	GI_cs; RusPer; Reviews; Tw_split; LJ_split; FB_split	Model 1	Seq. feat	0.86	0.03	0.27
17	RusPer; Reviews; Tw_split; LJ_split; FB_split	Model 1	Seq. feat	0.82	0.03	0.23
18	GI_cs	Model 1	Seq. feat	0.85	0.03	0.26
19	Tw_split, LJ_split, FB_split	Model 1	Seq. feat	0.81	0.03	0.22
20	GI_cs; RusPer; Reviews	Model 1	Seq. feat	0.87	0.03	0.28
21	RusPer; Reviews	Model 1	Seq. feat	0.79	0.02	0.2
22	GI_cs_a_b	Model 1	Seq. feat	0.84	0.03	0.25

Table 5: Results of experiments on *imbalanced* datasets.

'c' subsets) dominate over opposite-gender-mimicking samples ('b' subset), the network learns the straightforward gender-indicating morphological features (like genus in Russian). And, vice versa, if gender-deceptive samples dominate over actual-gender ones, the network still relies on the straightforward morphological features, just reverting them. So, by balancing the training set by the number of actual-gender and gender-deceptive samples, we aim to make the network ignore the straightforward gender-indicating features and infer the deeper, non-trivial ones.

5 Results

Tables 5, 6 demonstrates the F1 score of achieved models, trained on different sets of imbalanced and balanced training datasets respectively. The Model 1 gives the best result of 88% ± 3%, that is about 30% more than the bl. The best model based on the conventional Gradient Boosting algorithm gives the result of 70% ± 7%, which is 20% above bl. The efficiency of SVM model learning directly depends on the training set size is shown in Fig 1. The gain in F1 while the set size variates from 10% to 100% is about 12%.

6 Conclusion

The presented approach based on deep neural networks with CNN layers proved to be well-founded for solving the problem of identifying the gender of a text author. In our previous works, a neural network was presented on the basis of a combination of CNN and LSTM [12],

#	Train datasets	Model	Features	F1	F1_std	Delta *(bl 0.5)*
1	GI_cs_a_b	GB	TF-IDF	0.65	0.03	0.06
2	GI_cs_a	GB	TF-IDF	0.78	0.02	0.19
3	GI_cs_a	SVM	TF-IDF	0.73	0.02	0.14
4	GI_cs; RusPer; Reviews; Tw_split; LJ_split; FB_split.	Model 1	Seq. feat	0.88	0.03	0.38
5	RusPer; Reviews; Tw_split; LJ_split; FB_split;	Model 1	Seq. feat	0.84	0.02	0.34
6	GI_cs	Model 1	Seq. feat	0.87	0.03	0.37
7	Tw_split; LJ_split; FB_split;	Model 1	feature	0.82	0.02	0.32
8	Tw; LJ; FB	Model 1	Seq. feat	0.77	0.04	0.27
9	GI_cs; RusPer; Reviews	Model 1	Seq. feat	0.88	0.03	0.38
10	RusPer; Reviews	Model 1	Seq. feat	0.79	0.03	0.29
11	GI_cs_a_b	Model 1	Seq. feat	0.86	0.03	0.36
12	GI_cs_a	GB	TF-IDF	0.79	0.02	0.29

Table 6: Results of experiments on *balanced* datasets.

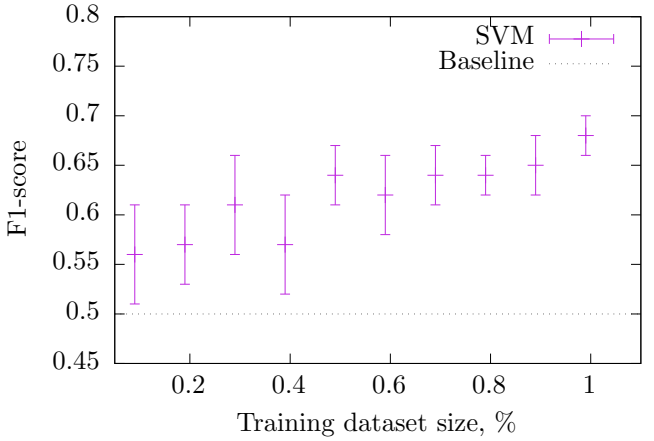

Figure 1: Dependence of estimations of SVM from the size of training set.

in which the achieved F1-score of 86% and the increase in comparison with bl of 23% correspond to the results obtained in this work. So, our results show that the state of the art in gender identification task with the RusProf corpus is about 88% with the superiority of deep learning models (Convolutional neural network - Model 1). Another consequence of the work is that the use of the collected corpus GI_cs allows to increase the accuracy of the task of identifying gender of non-deceptive Russian texts. This was proved by training on the Crowdsoursing corpus with testing on the GI corpus. So, this allows to exploit the same approach for texts with gender decepton in the future.

Acknowledgements

This research is supported by the Russian Science Foundation, project No 16-18-10050. This work has been carried out using computing resources of the federal collective usage center Complex for Simulation and Data Processing for Mega-science Facilities at NRC 'Kurchatov Institute', http://ckp.nrcki.ru/

References

[1] Francisco Rangel, Paolo Rosso, Martin Potthast, and Benno Stein. Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter. *Working Notes Papers of the CLEF*, 2017.

[2] Angelo Basile, Gareth Dwyer, Maria Medvedeva, Josine Rawee, Hessel Haagsma, and Malvina Nissim. N-gram: New groningen author-profiling model. *arXiv preprint arXiv:1707.03764*, 2017.

[3] Matej Martinc, Iza Škrjanec, Katja Zupan, and Senja Pollak. Pan 2017: Author profiling-gender and language variety prediction. *Cappellato et al.[13]*, 2017.

[4] Yasuhide Miura, Tomoki Taniguchi, Motoki Taniguchi, and Tomoko Ohkuma. Author profiling with word+ character neural attention network. *Cappellato et al.[13]*.

[5] Tatiana Litvinova, Pavel Seredin, Olga Litvinova, and Olga Zagorovskaya. Differences in type–token ratio and part-of-speech frequencies in male and female russian written texts. In *Proceedings of the Workshop on Stylistic Variation*, pages 69–73, 2017.

[6] Tatiana Litvinova, Olga Litvinlova, Olga Zagorovskaya, Pavel Seredin, Aleksandr Sboev, and Olga Romanchenko. "ruspersonality": A russian corpus for authorship profiling and deception detection. In *Intelligence, Social Media and Web (ISMW FRUCT), 2016 International FRUCT Conference on*, pages 1–7. IEEE, 2016.

[7] RusProfiling Lab. Rusprofiling corpus of russian texts. [online], 2017. `http://rusprofilinglab.ru/rusprofiling-at-pan/corpus/`.

[8] Andrey Kutuzov and Elizaveta Kuzmenko. Building web-interfaces for vector semantic models with the webvectors toolkit. *EACL 2017*, page 99, 2017.

[9] Nal Kalchbrenner, Ivo Danihelka, and Alex Graves. Grid long short-term memory. *arXiv preprint arXiv:1507.01526*, 2015.

[10] Eliyahu Kiperwasser and Yoav Goldberg. Simple and accurate dependency parsing using bidirectional lstm feature representations. *arXiv preprint arXiv:1603.04351*, 2016.

[11] Lin Wu, Chunhua Shen, and Anton van den Hengel. Personnet: Person re-identification with deep convolutional neural networks. *arXiv preprint arXiv:1601.07255*, 2016.

[12] Aleksandr Sboev, Tatiana Litvinova, Irina Voronina, Dmitry Gudovskikh, and Roman Rybka. Deep learning network models to categorize texts according to author's gender and to identify text sentiment. In *Computational Science and Computational Intelligence (CSCI), 2016 International Conference on*, pages 1101–1106. IEEE, 2016.