Classification of online toxic comments using the logistic regression and neural networks models

Cite as: AIP Conference Proceedings **2048**, 060011 (2018); https://doi.org/10.1063/1.5082126
Published Online: 11 December 2018

Mujahed A. Saif, Alexander N. Medvedev, Maxim A. Medvedev, and Todorka Atanasova

Classification of Online Toxic Comments Using the Logistic Regression and Neural Networks Models

Mujahed A. Saif[1, a)], Alexander N. Medvedev[1,2, b)], Maxim A. Medvedev[1,2, c)], Todorka Atanasova[3, d)]

[1]*Ural Federal University, Mira 19, Yekaterinburg, Russia, 620002*
[2]*Institute of Industrial Ecology UB RAS, Sophy Kovalevskoy 20, Yekaterinburg, Russia, 620990*
[3]*Varna University of Economics, 9002 Varna, Bulgaria*

[a)]m.a.saif@urfu.ru
[b)]Corresponding author: alnikmed52@gmail.com
[c)]medvedevmaa@gmail.com
[d)]t_atanasova@ue-varna.bg

Abstract: The paper addresses the questions of abusive content identification in the Internet. It is presented the solving of the task of toxic online comments classification, which was issued on the site of machine learning Kaggle (www.Kaggle.com) in March of 2018. Based on the analysis of initial data, four models for solving the task are proposed: logistic regression model and three neural networks models - convolutional neural network (Conv), long short-term memory (LSTM), and Conv + LSTM. All models are realized as a program in Python 3, which has simple structure and can be adapted to solve other tasks. The results of the classification problem solving with help of proposed models are presented. It is concluded that all models provide successful solving of the task, but the combined model Conv + LSTM is the most effective, so as it provides the best accuracy.

INTRODUCTION

The phenomenal development of computer science and communications has given us one of the greatest innovations of the 21st century "the Internet", where one can communicate with anyone on this planet having only two things: access to the Internet and a smart phone.

The actual start of the Internet was in 1990, when Tim Berners-Lee developed the first Web browser called World Wide Web [1]. At that time, communication between two people was accomplished via e-mail servers and it was filled with spam emails. To solve this problem many algorithms were developed, which successfully classified emails as spam or not spam [2]. Nowadays, the flow of data over the internet has grown dramatically, especially with the appearance of social networking sites. Due to this, an important task now is the development of algorithms to automatically classify the social networks content as "positive" or "negative", in order to prevent possible harm to the society.

In recent years there have been many cases in which authorities arrested some users of social sites because of negative (abusive) content of their personal pages. For example, one Man in Thailand was jailed for 35 years for insulting monarchy on Facebook (https://www.theguardian.com/world/2017/jun/09/man-jailed-for-35-years-in-thailand-for-insulting-monarchy-on-facebook), and the educational authority in the southern USA state of Mississippi did not find a better option than to expel a teacher from the Batesville Intermediate Primary School, because of a racist comment on her Facebook page about dark-skinned people (https://www.clarionledger.com/story/news/2017/09/20/mississippi-teacher-fired-after-racist-facebook-post/684264001/).

Hence, there is a great need to classify these posts and comments before they are published.

In this paper, the application of such machine learning methods as "logistic regression" and "neural network" is considered to solve the problem of text classification. Nowadays, these methods are widely used in economic,

Proceedings of the 44th International Conference on Applications of Mathematics in Engineering and Economics
AIP Conf. Proc. 2048, 060011-1–060011-5; https://doi.org/10.1063/1.5082126
Published by AIP Publishing. 978-0-7354-1774-8/$30.00

environmental, medical and other studies [3-6]. The task is solved on the example of data of the site Kaggle.com (machine learning site: https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge, 16.03.2018), which has released recently a competition on the classifying of online comments into six rankings and set a prize for top three contenders.

TEXT CLASSIFICATION PROBLEM

It is easy for human to classify images or text, but it is difficult for computers, which deal only with numbers, and to be more accurate, they process numbers in the form of electrical impulses. Therefore, any data must be converted into that form so computer can process it and give us back the result. The text classification algorithms use Natural Language Processing (NLP), Data Mining, and Machine Learning techniques to classify online comments [7-9]. Thus, before classifying, we have to analyze and vectorize the input data, and extract features from the text.

Looking at the data of Kaggle, we can see two csv files: "train.csv" and" test.csv". We use Python 3 software and its packages such as Pandas to upload these files, manipulate and examine the data (see Fig. 1-3).

```
>>> train_data.head()
                 id                                                comment_text  toxic  \
0  0000997932d777bf  Explanation\nWhy the edits made under my usern...      0
1  000103f0d9cfb60f  D'aww! He matches this background colour I'm s...      0
2  000113f07ec002fd  Hey man, I'm really not trying to edit war. It...      0
3  0001b41b1c6bb37e  "\nMore\nI can't make any real suggestions on ...      0
4  0001d958c54c6e35  You, sir, are my hero. Any chance you remember...      0

   severe_toxic  obscene  threat  insult  identity_hate
0             0        0       0       0              0
1             0        0       0       0              0
2             0        0       0       0              0
3             0        0       0       0              0
4             0        0       0       0              0
```

FIGURE 1. Result of "train_data.head".

```
>>> test_data.head()
                 id                                                comment_text
0  00001cee341fdb12  Yo bitch Ja Rule is more succesful then you'll...
1  0000247867823ef7  == From RfC == \n\n The title is fine as it is...
2  00013b17ad220c46  " \n\n == Sources == \n\n * Zawe Ashton on Lap...
3  00017563c3f7919a  :If you have a look back at the source, the in...
4  00017695ad8997eb             I don't anonymously edit articles at all.
>>>
```

FIGURE 2. Result of "test_data.head".

```
>>> train.columns.tolist()
['id', 'comment_text', 'toxic', 'severe_toxic', 'obscene', 'threat', 'insult', 'identity_hate']
>>> test.columns.tolist()
['id', 'comment_text']
>>> train = pd.read_csv('data/train.csv').fillna(' ')
>>> test = pd.read_csv('data/test.csv').fillna(' ')
>>> train.columns.tolist()
['id', 'comment_text', 'toxic', 'severe_toxic', 'obscene', 'threat', 'insult', 'identity_hate']
>>> test.columns.tolist()
['id', 'comment_text']
>>> train.shape
(159571, 8)
>>> test.shape
(153164, 2)
>>>
```

FIGURE 3. Basic information of the data.

The above figures show that the training data file has 8 columns and 159571 rows and test data file has two columns and 153164 rows. We are interested in the second column of both datasets and the last six columns of the training data set. To get more clear vision of what is in this datasets, the diagrams in Fig. 4 may be used.

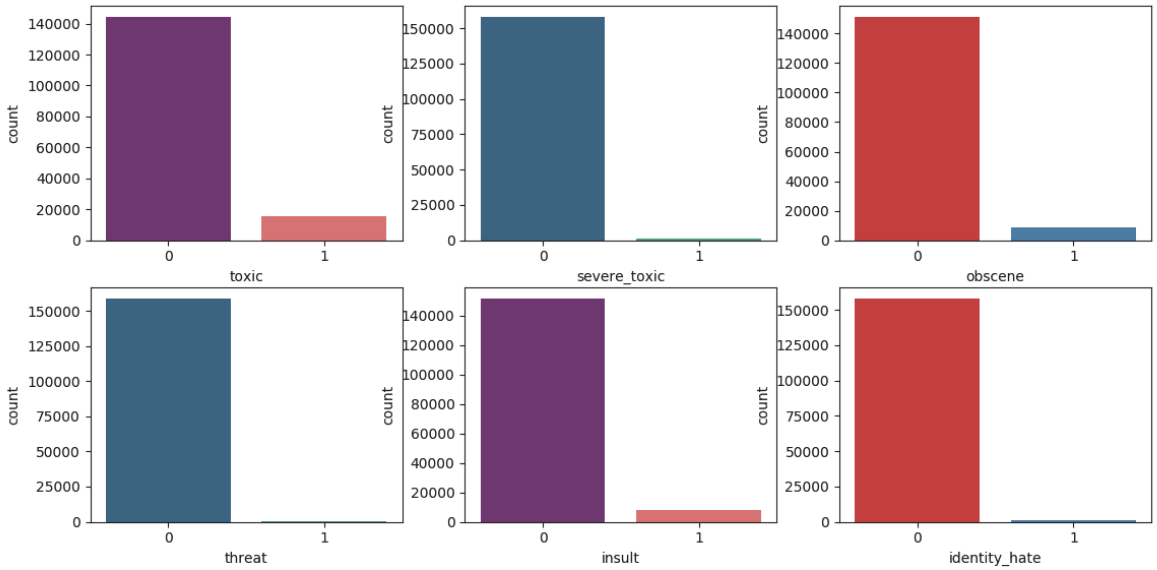


FIGURE 4. Data distribution on the classes.

From Fig. 4 it is obvious that most of the comments do not belong to any class; the "threat" and "identity hate" classes have the lowest number of comments; the most of comment does belong to "insulting" and "obscene" classes.

In order to solve the problem of classification we have to define some variables including X, Y:

class_names = ['toxic', 'severe_toxic', 'obscene', 'threat', 'insult', 'identity_hate'].

X_train, is the comment in the training dataset, X_test, is the comment in the test dataset, and Y_train the last six columns of the training dataset.

LOGISTIC REGRESSION MODEL

Since the Y_train has just two values 0 and 1, we are going to implement logistic regression model, but before that we are going to extract features from each comment using CountVectorizer from sklearn.feature_extraction.text. At first, we have to append the test data to the training data to vectorize the text and avoid any dimensional error. Then, we split it again to the original variables, but this time the type of data has changed from <class 'pandas.core.series.Series'> into <class 'scipy.sparse.csr.csr_matrix'>. Next, we call import LogisticRegression and fit X_train with Y_train.

The idea here is to implement logistic regression for each output column using the model with X_train, Y_train[class_names[i]]; then to make a prediction for X_test; then to store the result into the data set Y_test appending it to the test data and storing in a csv file.

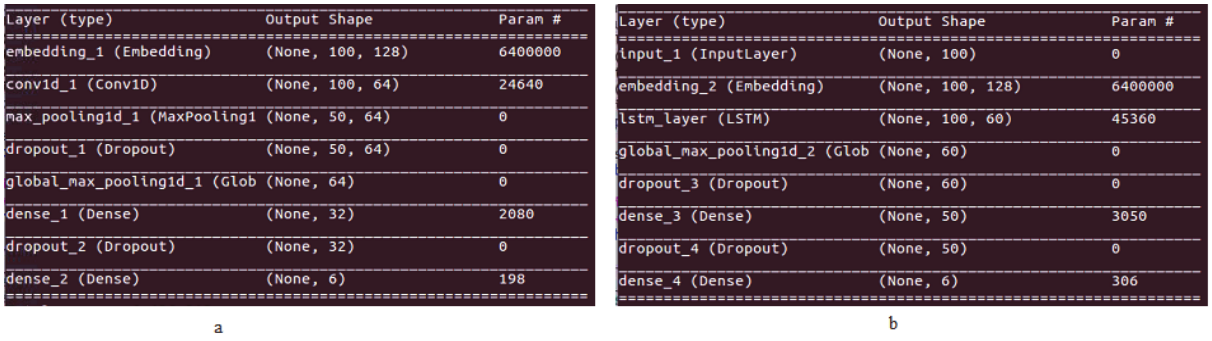
NEURAL NETWORK MODELS

The appearance of recurrent neural network (RNN) and long-term short-term memory (LSTM) has provided great progress in the processing of multimedia data and produced the most advanced results in speech recognition, digital signal processing, video processing and text data analysis. The RNN architecture provides a tool for processing and searching for hidden patterns in data such as text [10]. The layer LSTM are especially designed to

avoid the long-term dependency problem in neural network. Remembering of information for long periods of time is practically their default behavior, not something they could be taught.

Convolutional neural network (CNN) - the special architecture of artificial neural networks, proposed by Jan Lekun in 1988 and aimed at effective image recognition, is part of deep learning technologies. The network uses some features of the human visual cortex, in which so-called simple cells reacting to straight lines from different angles were discovered, and complex cells whose reaction is associated with the activation of a certain set of simple cells. Thus, the idea of convolutional neural networks consists in the alternation of convolution layers and pooling layers. The network structure is unidirectional (without feedbacks), essentially multilayered. For training, standard methods are used, most often the method of back propagation of the error. The function of activation of neurons (transfer function) may be any, at the choice of the researcher.

The third model is a combination of RNN and CNN and all models have an embedding layer. A word embedding is a class of approaches for representing words and documents using a dense vector representation where a vector represents the projection of the word into a continuous vector space. It would take three parameters max_features, embed_size, input_length of 50000,128,100 respectively. The architecture of LSTM and RNN can be seen in Fig. 5, and they always end up within output shape of 6 since we have six classes.



Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 100, 128)	6400000
conv1d_1 (Conv1D)	(None, 100, 64)	24640
max_pooling1d_1 (MaxPooling1	(None, 50, 64)	0
dropout_1 (Dropout)	(None, 50, 64)	0
global_max_pooling1d_1 (Glob	(None, 64)	0
dense_1 (Dense)	(None, 32)	2080
dropout_2 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 6)	198

a

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 100)	0
embedding_2 (Embedding)	(None, 100, 128)	6400000
lstm_layer (LSTM)	(None, 100, 60)	45360
global_max_pooling1d_2 (Glob	(None, 60)	0
dropout_3 (Dropout)	(None, 60)	0
dense_3 (Dense)	(None, 50)	3050
dropout_4 (Dropout)	(None, 50)	0
dense_4 (Dense)	(None, 6)	306

b

FIGURE 5. The architecture of used CNN (a) and RNN (b) networks.

RESULTS AND DISCUSSION

A 0.1 of the training data set was used (that is 15958 comments) to estimate the accuracy of the algorithm in both methods and obtained the values of 0.91 and 0.9820 using logistic regression and LSTM + conv, respectively. Table 1 shows the comparison of the accuracy between the three types of neural network models tested on 0.33 of the training data. The best model in this paper, which contains 2 LSTM layers and 4 conv layers, has has got a score of 0.9645. On the information of the site Kaggle.com, the first place in the competition has a model with score of 0.985. So, the results obtained here can be improved that can be done, in particular, by the varying of such parameters as maxfeatures and maxlen, which are experimental ones.

TABLE 1. Comparing the validation results of the neural network on 0.33 of the training data.

Network type	Layers	Loss	Acc	Val_loss	Val_acc
LSTM+CONV	(2+4)	0.0423	0.9839	0.0522	0.9816
CONV layer	4	0.0268	0.9893	0.0585	0.9806
LSTM layer	1	0.2117	0.9680	0.2228	0.9639

The considered models are represented as a script in Python 3, which is in free access (https://github.com/aldkak/toxic_text_classification) and may be used to extract comments from social sites and classify them according to the requirements of users.

CONCLUSION

In the paper, four models for online abusive comments classification are proposed, which are: logistic regression model and three neural networks models - convolutional neural network (Conv), long short-term memory (LSTM), and Conv + LSTM. Based on the obtained results, one can conclude that the most effective is the combined model Conv + LSTM, which provides the best accuracy: 0.9820 and 0.9645 when testing on 0.1 and 0.33 of the training data set respectively. All models are realized as a script in Python 3, which is in free access and may be used to extract comments from social sites and classify them according to the requirements of users.

REFERENCES

[1] Tim Berners-Lee, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen, Arthur Secret, The world-wide web, available at https://www.sciencedirect.com/science/article/pii/B9780080515748500959.

[2] Spam filtering of e-mail messages based on neural network and neuro models, available at https://cyberleninka.ru/article/n/spam-filtratsiya-elektronnyh-pochtovyh-soobscheniy-na-osnove-neyrosetevoy-i-neyronechetkoy-modeley.

[3] O.I. Nikonov, M.A. Medvedeva, F.P. Chernavin, "Using the Committee Machine Method to Forecasting on the FOREX", *Proceedings of Second International Conference on Mathematics and Computers in Sciences and in Industry (MCSI 2015)* (Conference Publishing Services (CPS), 2015), pp. 240-243.

[4] Oleg I. Nikonov, Fedor P. Chernavin, and Marina A. Medvedeva, "The problems of classification: Method of committees", AIP Conf. Proc. 1738, 110004 (2016).

[5] A. Medvedev, D. Taushankova, M. Medvedeva, A. Varaksin, A. Sergeev, "Snow Pollution Regression Model for Karabash City of Russia", *14th GeoConference on Ecology, Economics, Education and Legislation. SGEM2014, Conference Proceedings* (2014), Volume 1, pp. 601-606.

[6] Anastasia Chirkina, Marina Medvedeva, and Evgeny Komotskiy, "CIN classification and prediction using machine learning methods", AIP Conf. Proc. 1836, 020010 (2017).

[7] M. Ikonomakis, S. Kotsiantis, V. Tampakas, "Text Classification Using Machine Learning Techniques", WSEAS Transactions on Computers, Issue 8, Volume 4, pp. 966-974 (August 2005).

[8] Siwei Lai, Liheng Xu, Kang Liu, Jun Zhao, "Recurrent Convolutional Neural Networks for Text Classification", *AAAI'15 Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, Texas, January 25 - 30, 2015* (AAAI Press, 2015), pp. 2267-2273.

[9] Zhao Jianqiang, Gui Xiaolin, "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis", IEEE Access 5, 2870-2879 (2017).

[10] A. Ullah et al., "Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features", IEEE Access 6, 1155-1166 (2018).