JEDIC Journal of Ethics and Diversity in International Communication

| e-ISSN: 2792-4017 | www.openaccessjournals.eu | Volume: 1 Issue: 5

Using Conventional Neural Networks for the Problem of Text Classification

Dildora Kabulovna Muhamediyeva, Nigora Nurmaxamadovna Abdurakhmanova Nilufar, Sirojidinovna Mirzayeva

Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

Abstract.

Convolutional neural networks are a powerful machine learning tool that aims to efficiently recognize and classify images. The success of using convolutional neural networks for images has spawned many attempts to use this tool in other tasks. In this paper, the main methods of using convolutional neural networks for the text classification problem are investigated. Experiments on large text data have been carried out, showing that convolutional neural networks for the text classification problem can achieve a quality similar to or better than traditional methods.

Introduction

Задача классификации текстов становится все более актуальной в связи с постоянно растущим объемом информации в интернете и потребностью в ней ориентироваться. К примеру, классификация текстов необходима для решения следующих задач:

1. Fighting spam.

Spam is unsolicited mailings that may be sent to an email address. They may contain promotional offers or computer viruses. The task of fighting spam is to classify all emails into two classes: spam and non-spam.

2. Recognition of the emotional coloring of texts.

The task is to assess the opinion of the author in relation to objects, for example, based on reviews of these objects. Often such a task must be solved in order to issue relevant recommendations.

3. Division of sites into subject catalogs.

This problem is solved by search engines and involves processing documents and assigning them to one of several categories, the list of which is predefined.

4. Personification of advertising.

Contextual advertising is the main source of income for IT companies. It is displayed to website visitors whose area of interest potentially coincides or overlaps with the subject of the advertised product or service, the target audience, which increases the likelihood of their response to advertising. The area of interest is determined by the text of the Internet pages viewed by the user.

Due to the importance of this task, many machine learning competitions with valuable prizes are being held to solve it, new methods are being explored to achieve better classification quality.

2. Statement of the classification problem

Classification solves the following problem. A finite set of classes is given and there is a set of objects, for a finite subset of which it is known to which class they belong. This subset is called the training sample. The class of the remaining objects is unknown. It is required to construct an algorithm capable of classifying an arbitrary object from the original set.

To classify an object means to indicate the number (or name of the class) to which this object belongs. In text classification tasks, objects are text documents. Let us write down the formal statement of the text classification problem.

$D = \{d_1, .., d_n\}$- set of text documents. Each document$d \in D$is a sequence of words.

$Y = \{y_1, .., y_n\}$- finite set of class labels.

ISSN 2792-4017 (online), Published under Volume: 1 Issue: 5 in October-2021

Traditional machine learning methods for text classification

Typically, the text classification problem is solved using the following steps:

1. Text preprocessing.

2. Translation of texts into the real space of features, where each document will be associated with a vector of fixed length.

3. Selection of a machine learning algorithm for classification.

Text preprocessing

All natural language texts have a large number of words that do not carry information about the given text. For example, in English, such words are articles; in Russian, they include prepositions, conjunctions, particles. These words are called noise or stop words. To achieve a better quality of classification at the first stage of text preprocessing, it is usually necessary to delete such words. The second stage of text preprocessing is bringing each word to a stem that is the same for all its grammatical forms. This is necessary, since words carrying the same meaning can be written in different forms. For example, the same word can occur in different declensions, have different prefixes and endings.

Translating text into vector representation

Most modern machine learning algorithms are focused on the feature description of objects, so all documents are usually translated into the real feature space. To do this, they use the idea that words are responsible for the belonging of a document to a certain class, and texts from one class will use many similar words. 7 The most well-known methods for translating a text into a feature space are based on statistical information about words. When using them, each object is converted into a vector, the length of which is equal to the number of words used in all texts of the sample.

Bag of Words

Bag of Words is a model for translating text into a vector representation. The main assumption of this method is that the word order in the document is not important, and the collection of documents can be viewed as a simple selection of document-word pairs$(d, w)$where$d \in D$, W$\in W_d$.

Bag of Ngrams & TF IDF

Often information in the text is carried not only by individual words, but also by a certain sequence of words. For example, phraseological units are stable combinations of words whose meaning is not determined by the meaning of the words included in them, taken separately. For example, the phrase "Like a fish in water" means to feel confident, to be very good at something. The meaning of this expression will be conveyed incorrectly if you take into account its words separately. In order to take into account such features of the language, it is proposed to take into account, in addition to words, N-grams when translating texts into a vector representation. N-grams are sequences of N words. For example, for the text "mama soap frame" we get the bigrams "mama soap" and "soap frame". In the problem of text classification, N-grams are indicators that the data of N words have met side by side.The Bag of Ngrams & TF IDF method is similar to the Bag of Words & TF IDF method, only the feature vector for each document, in addition to TF IDF words, contains TF IDF of all sequences from N words.

Choice of classification algorithm

The resulting feature space in this method will be very sparse and will have a high dimension due to the fact that there are usually many different words found in the entire sample. Because of this, linear machine learning methods are most often used for this task.

Convolutional neural networks

With the advent of large amounts of data and great computational capabilities, neural networks are actively used. Convolutional neural networks, the architecture of which was proposed by Jan Lekun [12] and aimed at efficient image recognition, have gained particular popularity.The network architecture got its name due to the presence of the convolution operation, the essence of which is that each fragment of the image is multiplied by the convolution matrix (kernel) element by element, and the result is summed up and written to a similar position in the output image. The network architecture is based on a priori knowledge from the subject area of computer vision: an image pixel is more strongly associated with a neighboring one (local correlation) and an object in the image can be found in any part of the image. Light neural networks received special attention after the ImageNet competition, which took place in October 2012 and was devoted to the classification of objects in photographs. The competition required pattern recognition in

ISSN 2792-4017 (online), Published under Volume: 1 Issue: 5 in October-2021

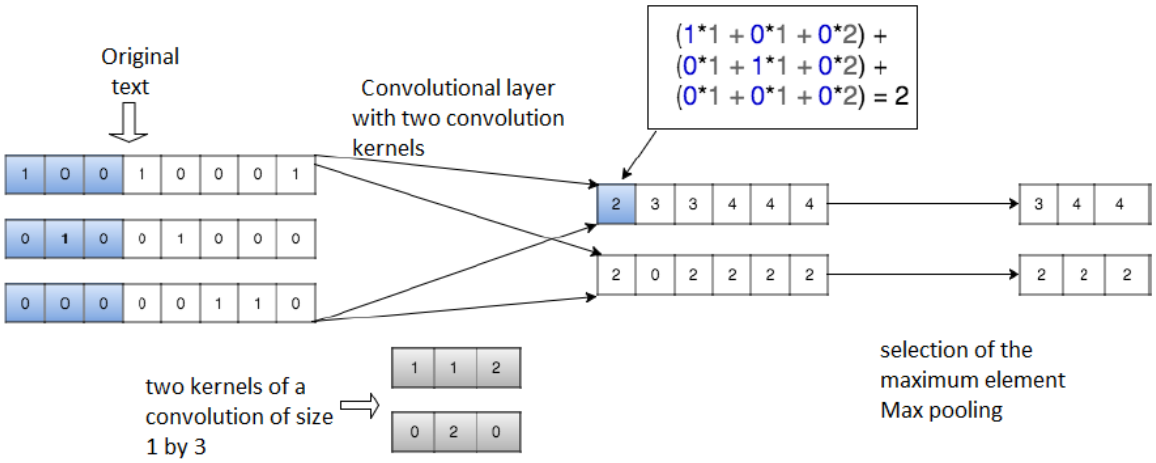
1000 categories. The winner of this competition - Alex Krizhevsky, using a convolutional neural network, significantly surpassed the rest of the participants [6]. The success of applying light neural networks to image classification has led to many attempts to apply this method to other problems. Recently, they have been actively used for the task of text classification.

Models of using convolutional neural networks for text classification

This section will describe the main approaches to using convolutional neural networks for the problem of text classification.

Symbol-by-symbol approach

Symbol-by-symbol approach for text classification using convolutional neural networks was suggested in the article. Let's describe this method in more detail. Let's call an ordered set of symbols an alphabet. Let the chosen alphabet consist of m symbols. Each character of the alphabet in the text is encoded using 1 - m - encoding. (i.e., each character will be associated with a vector of length m whose element is equal to one, in a position equal to the ordinal number of the character in the alphabet, and zero in all other positions.) If a character is encountered in the text that is not included in the alphabet, then it is necessary to encode it a vector of length m consisting of all zeros. The first `characters are selected from the text. The l parameter must be large so that the first l characters contain enough information to determine the class of the entire text**.**



1-fig. Symbol-by-symbol approach

Let us describe this approach formally.

Let xi — vector of the i-th character in the text.

$$x_{1:l=}x_1 \oplus x_2 \oplus \ldots \oplus x_l$$

Here⊕-vector combining operation.

Convolutional layer:

$$c_i = f(w \cdot x_{i:i+h-1} + b)$$

c=$[c_1, c_2, \ldots, c_{n-h+1}]$

f - neural network activation function

b – constant.

Word Encoding Approach

The approach was described in article [1]. In this approach, each word in the text is associated with a vector of fixed length, then a matrix is compiled from the obtained vectors for each sample object, which, similarly to images, is fed to

ISSN 2792-4017 (online), Published under Volume: 1 Issue: 5 in October-2021

the input of a convolutional neural network. In Fig. 2 shows an example of a convolutional neural network using word coding.

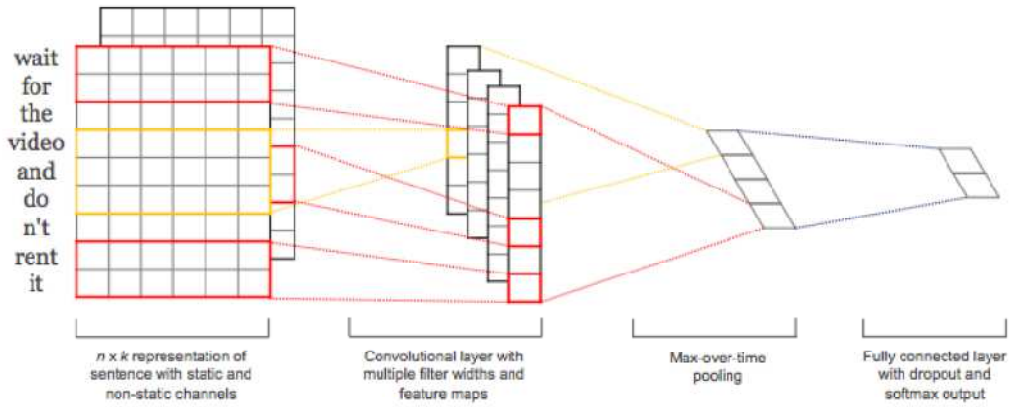


Fig 2. Symbol by symbol approach

Данныедляобучения

For training, we choose a corpus of short texts. It contains 114,991 positive, 111,923 negative tweets, and a database of 17,639,674 untagged tweets.

Before starting the training, the texts went through a preprocessing procedure:

• converting to lower case;

• removing punctuation marks.

Next, we split the dataset into training and test sets in a 4: 1 ratio.

- from sklearn.model_selection import train_test_split
-
- x_train, x_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=1)

ISSN 2792-4017 (online), Published under Volume: 1 Issue: 5 in October-2021

JEDIC
Journal of Ethics and Diversity in International Communication
| e-ISSN: 2792-4017 | www.openaccessjournals.eu | Volume: 1 Issue: 5

Vector display of texts

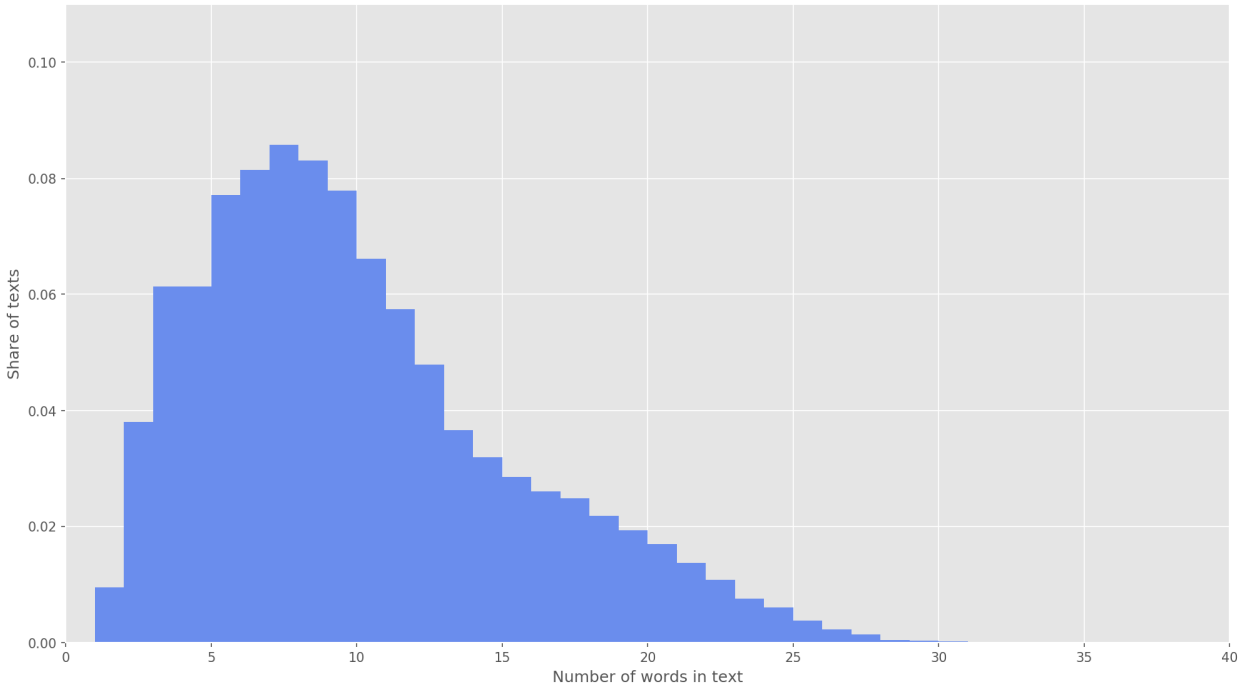


Fig 3. Distribution of text lengths.

At the next stage, each text was mapped into an array of token identifiers. I chose the dimension of the text vector s = 26, since this value completely covers 99.71% of all texts in the generated corpus (Fig. 3). If, during analysis, the number of words in a tweet exceeded the height of the matrix, the remaining words were discarded and not included in the classification. The final dimension of the proposal matrix was s × d = 26 × 200.

```
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
SENTENCE_LENGTH = 26
NUM = 100000
defget_sequences(tokenizer, x):
sequences = tokenizer.texts_to_sequences(x)
return pad_sequences(sequences, maxlen=SENTENCE_LENGTH)
# Create and train a tokenizer
tokenizer = Tokenizer(num_words=NUM) tokenizer.fit_on_texts(x_train)
# Map each text to an array of token IDs
x_train_seq = get_sequences(tokenizer, x_train)
x_test_seq = get_sequences(tokenizer, x_test)
```

To build a neural network, we used the Keras library, which acts as a high-level superstructure over TensorFlow, CNTK and Theano. Keras has excellent documentation and a blog that covers many machine learning tasks, such as

ISSN 2792-4017 (online), Published under Volume: 1 Issue: 5 in October-2021

initializing an embedding layer. In our case, the embedding layer was initiated by the weights obtained from training Word2Vec. To minimize changes to the embedding layer, I froze it in the first step of training.

```
from keras.layers import Input
```

```
from keras.layers.embeddings import Embedding
```

```
tweet_input = Input(shape=(SENTENCE_LENGTH,), dtype='int32')
```

```
tweet_encoder = Embedding(NUM, DIM, input_length=SENTENCE_LENGTH,
```

```
weights=[embedding_matrix], trainable=False)(tweet_input)
```

The developed architecture uses filters with height h = (2, 3, 4, 5), which are designed for parallel processing of bigrams, trigrams, 4-grams and 5-grams, respectively. Added to the neural network 10 convolutional layers for each filter height, the activation function is ReLU. Recommendations for finding the optimal height and number of filters can be found in [2].

After processing with convolution layers, feature maps were sent to subsampling layers, where the 1-max-pooling operation was applied to them, thereby extracting the most significant n-grams from text. At the next stage, the merging into a common feature vector (merging layer) took place, which was fed into a hidden fully connected layer with 30 neurons. At the last stage, the final feature map was fed to the output layer of the neural network with a sigmoidal activation function.

Since neural networks are prone to overfitting, after the embedding layer and in front of the hidden fully connected layer, I added dropout regularization with a vertex outlier probability p = 0.2.

```
from keras import optimizers
```

```
from keras.layers import Dense, concatenate, Activation, Dropout
```

```
from keras.models import Model from keras.layers.convolutional import Conv1D
```

```
from keras.layers.pooling import GlobalMaxPooling1D branches = []
```

```
# Adding dropout regularization
```

```
x = Dropout(0.2)(tweet_encoder)
```

```
for size, filters_count in [(2, 10), (3, 10), (4, 10), (5, 10)]:
```

```
for i in range(filters_count):
```

```
# Add a convolution layer
```

```
branch = Conv1D(filters=1, kernel_size=size, padding='valid', activation='relu')(x)
```

```
# Add a downsampling layer
```

```
branch = GlobalMaxPooling1D()(branch)
```

ISSN 2792-4017 (online), Published under Volume: 1 Issue: 5 in October-2021

```
branches.append(branch)
# Concatenate feature maps
x = concatenate(branches, axis=1)
# Add dropout regularization
x = Dropout(0.2)(x)
x = Dense(30, activation='relu')(x)
x = Dense(1)(x)
output = Activation('sigmoid')(x)
model = Model(inputs=[tweet_input], outputs=[output])
```

The result of the program

```
Corpus length: 562404
Number of sequences: 187448
Unique characters: 69
Vectorization...
epoch 1
1465/1465 [==============================] - 11s 7ms/step - loss: 2.2591
epoch 2
1465/1465 [==============================] - 10s 7ms/step - loss: 1.6166
epoch 3
1465/1465 [==============================] - 9s 6ms/step - loss: 1.5189
epoch 4
1465/1465 [==============================] - 10s 6ms/step - loss: 1.4653
epoch 5
1465/1465 [==============================] - 10s 7ms/step - loss: 1.4322
epoch 6
1465/1465 [==============================] - 10s 7ms/step - loss: 1.4081
epoch 7
1465/1465 [==============================] - 10s 7ms/step - loss: 1.3902
epoch 8
1465/1465 [==============================] - 9s 6ms/step - loss: 1.3739
epoch 9
1465/1465 [==============================] - 10s 7ms/step - loss: 1.3612
epoch 10
1465/1465 [==============================] - 10s 7ms/step - loss: 1.3512
--- Generating with seed: "ngs in
baker street, buried among his old books, and alterna"
------ temperature: 0.2
ngs in
baker street, buried among his old books, and alternated the station of the started and strongelves which have been the more silent of the startion of the propirer of the strange and a companion.
derable that i have a surprised the busing of my friend of the descapters with a strongely and before that we were miss dather in the day. i have never instantly most case the commisoros men. his
as has had been success that i pups and sat down the windows. i should do you
mey of my destantered. if you rashed moghisting in
remarks down his heading racem of his inually before i saw that there was out in their steps
```

Conclusion

This paper discusses the main methods of text classification, as well as the recently proposed character-by-character approach using convolutional neural networks. The paper implements a convolutional neural network with a character-by-character approach that classifies texts and shows that on large data, the character-by-character approach shows a better quality of classification in comparison with traditional methods. The main contribution of this work is the study of the influence of text preprocessing on the quality of classification using convolutional neural networks with a character-by-character approach.

References

1. Muhamediyeva D.K., Nurumova A.Yu., Muminov S.Yu. Study Of Multicomponent Cross-Diffusion Systems Of Biological Population With Convective Transfer // European Journal of Molecular & Clinical Medicine ISSN 2515-8260 Volume 7, Issue 11, 2020, pp. 2934-2944.
2. Cliche M. BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs //Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). — 2017. — C. 573-580.
3. Mikolov T. et al. Distributed Representations of Words and Phrases and Their Compositionality //Advances in Neural Information Processing Systems. — 2013. — C. 3111-3119.

ISSN 2792-4017 (online), Published under Volume: 1 Issue: 5 in October-2021