

Text Tonality Classification Using a Hybrid Convolutional Neural Network with Parallel and Sequential Connections Between Layers

Roman Peleshchak¹, Vasyl Lytvyn¹, Ivan Peleshchak¹, Andriy Khudyy¹, Zoriana Rybchak¹, Solomiya Mushasta¹

¹ Lviv Polytechnic National University, 12 Stepana Bandera Street, Lviv, 79000, Ukraine

Abstract

The analysis of the tonality of texts is an urgent problem in the field of natural language processing, which is often solved with the help of convolutional neural networks. However, most of these CNN models focus only on the study of local functions, ignoring global features. In this paper, a hybrid convolutional neural network with parallel-sequential connections between layers and from the layer of maximum pulling obtained from the matrix of the original text is proposed for the analysis of text tonality. The proposed hybrid convolutional neural network extracts text features using a parallel-connected convolutional block. Then the neural network classifies the features and combines these features with the original text features. The model of the proposed neural network is able to study both local and global features of short texts and has less convergence time and computing resource compared to the parallel DenseNet. Hybrid convolutional neural network with parallel-sequential connections between layers has a higher efficiency of text tone classification in 6 different databases compared to the base models CNN, TextCNN, FastText, DPCNN.

Keywords

Text tonality, classification, convolutional neural network.

1. Introduction

The challenges of natural language processing are becoming increasingly important due to the ever-increasing amount of information on the Internet and the need to navigate this information. Tasks that are widely used in natural language processing include text classification, creating chatbots or generating answers to user questions, machine translation from one language to another, language recognition, spelling, identifying parts of speech in a sentence and their annotation, rewriting text information for creating web content. The labeled data set which contains text documents and their labels is used to train the classifier.

Text classification is widely used in tonal analysis (Imdb, Yelp review classification), stock market analysis, for automated e-mail responses. Methods based on in-depth learning of neural networks have become current practice along with classical algorithms for text mining. The following neural network architectures are used to solve text classification problems: recurrent neural network, hierarchical attention network and convolutional neural network [1].

Document classification is a process of assigning documents to a certain category depending on their content. Text classification is necessary to solve the following tasks: personification in advertising; sites division by thematic catalogs; fighting against deceptive or misleading advertising correspondence (spam); text tone recognition, i.e. determining the color of emotions in the text.

COLINS-2022: 6th International Conference on Computational Linguistics and Intelligent Systems, May 12–13, 2022, Gliwice, Poland
EMAIL: rpele@ukr.net (R. Peleshchak); vasy117.lytvyn@gmail.com (V. Lytvyn); peleshchakivan@gmail.com (I. Peleshchak); Khudyy@ukr.net (A. Khudyy); zozyka3@gmail.com (Z. Rybchak); solomiyanytrebych@gmail.com (S. Mushasta)
ORCID: 0000-0002-0536-3252 (R. Peleshchak); 0000-0002-9676-0180 (V. Lytvyn); 0000-0002-7481-8628 (I. Peleshchak); 0000-0003-2029-7270 (A. Khudyy); 0000-0002-5986-4618 (Z. Rybchak); 0000-0003-4932-4113 (S. Mushasta)



© 2022 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

The aim of the work is to develop a new model of the architecture of a convolutional neural network with parallel and sequential connections between layers to classify the tonality of texts with increased efficiency.

2. Literature review

Most modern machine learning algorithms focus on describing the features of objects, so all documents are converted into a real feature space. There is an idea that words are responsible for the belonging of a document to a certain class, and the texts of one class will have many similar words. The best-known ways to transform text into feature space are based on statistical information about words. Each object (text) is converted into a vector, the length of which is equal to the number of words in all sample texts.

There are three main strategies for detecting features in the text tonality analysis: the Bag of Words model [1], the word embedding model [2, 3] and the graph network model [4]. Fig. 1 presents the structure of the study of the text tonality analysis in accordance with the three main methods used to identify features.

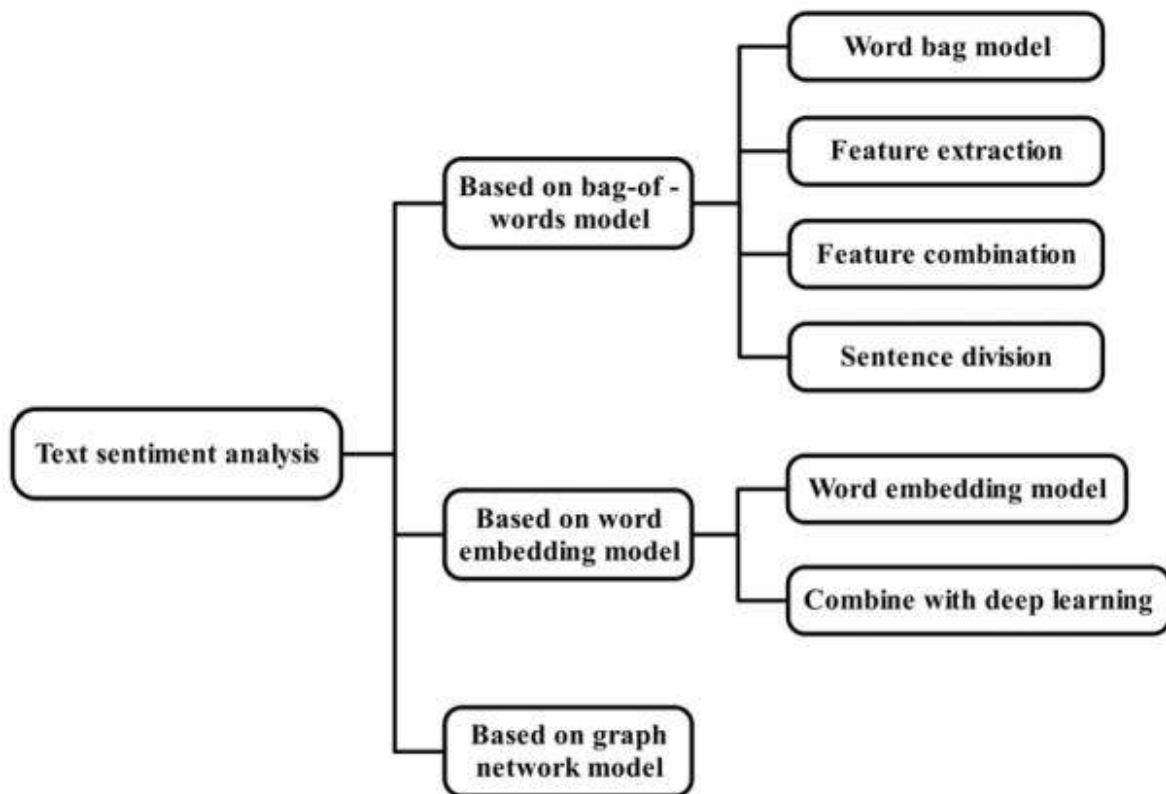


Figure 1: Text tonality analysis in accordance with the three main methods used to highlight the features

The main assumption of the Bag-of-words model is that the order of the words in the document is not important, and the collection of documents can be considered as a simple selection of pairs "document - word" (d, w) , where $d \in D$, where $D = \{d_1, \dots, d_n\}$ – set of text documents; $w \in W_d$, where $W_d = (w_1, \dots, w_{n_d})$ – sequence of words, n_d – length of the document d . All documents are presented in the form of a matrix $T = (t)_{d,w}$, each line of which corresponds to a certain document or text, and each column corresponds to a certain word. Element $t_{d,w}$ corresponds to the number of occurrences of the word w in document d .

Let's assume that a sentence is a syntactically ordered set of words. It consists of m words. Each word is encoded with 1 in the sentence – m coding, i.e. each word w will correspond to a vector of length m . The component of this vector, which corresponds to the encoded word, is equal to 1, in the

position that is equal to the ordinal number of the word placement in the sentence, and 0 – in all other positions. Therefore, the bag-of-words model was developed in the methods of detecting features, such as part-of-speech tagging, POS tagging, POST and n-grams tagging phrases.

Tagging a part of speech is also known as grammatical tagging or identification of parts of speech. It is the process of detecting a word in a text by belonging to a certain part of speech, based on its definition and context – on its connection with related words in a phrase, sentence, or paragraph.

The most popular way to convert text into a vector is the Bag-of-words & TF-IDF model [5]. As in the Bag-of-words model, all documents are presented in the form of a matrix $T = (t)_{d,w}$. But the element $t_{d,w}$ corresponds to the function TF IDF (w, d, D) of the word $w \in W_d$ in the document $d \in D$

Definitions 1. *TF-IDF – it is a statistical measure used to evaluate how important a word is in the context of a document. It is calculated by the formula:*

$$TF - IDF(w, d, D) = TF(w, d) \times IDF(w, D). \quad (1)$$

where TF – word frequency, which evaluates the importance of the word w_i within a certain document

$$TF(w, d) = \frac{n_i}{\sum_{k=1}^m n_k}. \quad (2)$$

n_i – the occurrences of the word i in the document.

$\sum_{k=1}^m n_k$ – the total number of words in the document.

IDF – the inverse document frequency. IDF consideration reduces the importance of commonly used words.

$$IDF(w, |D|) = \log \frac{|D|}{|d_i \supset w_i|}. \quad (3)$$

$|D|$ – number of documents in the corpus.

$|d_i \supset w_i|$ – the number of documents in which the word w_i occurs.

Often the information in the text is marked not only by individual words, but also by a sequence of words, i.e. phrases and phraseological units. In this case, the model of Bag of Ngrams & TF-IDF is used to get the language features when converting text into vector. N-grams [6] are sequences of N words in which a single word depends on several others. N-gram is an indicator that the data of N words related in the text classification tasks.

The Bag of N-grams & TF-IDF model is similar to the Bag-of-words & TF-IDF model, only the feature vector for each document except TF-IDF words contains TF-IDF of all N-word sequences.

$$TF - IDF(w, d, D, N) = TF(w, d) \times IDF(w, D) \cup TF(N, d) \times IDF(N, D). \quad (4)$$

$$TF(N, d) \times IDF(N, D) = \frac{N_g}{\sum_{g=1}^M N_g} \times \log \frac{|D|}{|d_g \supset P_g|}. \quad (5)$$

N_g – occurrences of g, N-gram in document.

$\sum_{g=1}^M N_g$ – the total number of N-gram in document; in case $M < m$.

$|d_g \supset P_g|$ – the number of documents in which N-gram P_g occurs.

POS tagging and N-gram were combined in [7, 8]. This model can improve the accuracy of classification according to experimental results. However, while analyzing the tonality of short texts [9, 10] it was found that this model cannot achieve satisfactory accuracy because in short texts the composition of sentences is quite arbitrary. Therefore, it is impractical to apply tagging of parts of speech to their analysis [9].

There are three main ways of analyzing the tone of short texts. They are based on vocabulary, traditional machine learning and deep learning. The model of text tonality analysis based on vocabulary allows to obtain the sentimental tendency of texts by calculating and evaluations of texts by words with sentimental information. The model is based on traditional machine learning, is independent of vocabulary and has the ability to independently study the sentimental features of texts. The model of text sentiments analysis is based on deep learning, allows more advanced and unspeakable sentimental features of texts [11]. Therefore, the features obtained by the model of text tonality analysis based on deep learning, are abstract and difficult to be expressed clearly.

In terms of differentiation in a sentence, the authors of [12] have developed a classification system that can identify words that convey sentimental polarity. The authors [13] used a classification algorithm developed using SentiWordNet emotion assessment and achieved significant performance improvements on six assessment datasets. SentiWordNet is an opinion lexicon that assigns three mood assessments to each WordNet syntax: positivity, negativity, and objectivity. Some studies have shown that using SentiWordNet for mood assessment of a word and adding it as a function can improve the accuracy of key analysis [7, 14]. The above given literature review shows that the text classification based on the bag-of-words model can achieve better results if the characteristics of the word are properly obtained.

However, the bag-of-words model has also some drawbacks, as it does not take into account the order of words in a sentence, i.e. syntax, and cannot convey deep-root semantic features and semantic combinations.

Text tonality analysis based on the word embedding model solves the problem of the bag-of-words model, if we apply the vector of words in the multidimensional feature space [15].

The methods of presenting words using a fixed length vector (when its length is equal to the number of words used in the sample) are the most well-known [16]. Each vector consists of zeros 0 and ones 1.

Word2Vec is a technology [17] focused on statistical processing of large arrays of textual information. Word2Vec collects statistics on the words occurrence in the data, removes the least and / or most common words, then solve the problem of dimensionality reduction the by using neural network methods to and produces compact vector representations of words of predetermined length.

In this case, Word2Vec maximizes the cosine similarity between word vectors that occur in similar contexts and minimize the cosine similarity between words that do not occur together.

A cosine similarity measures the similarity between two vectors. The cosine similarity between vectors \bar{A} and \bar{B} is calculated by the formula:

$$similarity = \cos(\theta) = \frac{\bar{A} \cdot \bar{B}}{\|\bar{A}\| \cdot \|\bar{B}\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}. \quad (6)$$

It should be noted that two different neural network architectures can be used for implementing Word2Vec technology to convert the word into a vector: Continuous Bag of Words and Skipgram.

The word embedding model is based on the principle of "distance similarity" and has a smoothing function.

Another advantage of the word embedding model is that it is a method of learning without a teacher. It is proved that the word embedding model can get more semantic and grammatical features than the bag-of-words model. This advantage allows word embedding model to achieve very good results in a variety of natural language processing tasks; [2, 15] developed the QVEC method to measure the evaluation of the effectiveness of the representation of the characteristics of different models of text analysis. The findings show that for 300D QVEC word vector assessment of the text tonality based on the word embedding model is higher than in other models. In recent years, a combination of word embedding model and in-depth learning model has been used for text analysis with better performance. The authors [18] developed a word embedding learning algorithm that combines word vectors with RNN and can be well applied to speech recognition. The authors [19, 20] combine word vectors with long-term short-term memory (LSTM) to achieve better efficiency. Although CNN, developed by the authors [1], has only one convolutional layer, its classification

efficiency is much better than in the conventional machine learning classification algorithm. However, this method cannot highlight convolutions in large texts. In 2017, the authors [21] identified dependencies in large texts by deepening the network. The authors [22] presented a structure similar to DenseNet, using abbreviations between the upper and lower convolution blocks, so that larger objects can be obtained from smaller feature combinations. However, the model used a convoluted core of a certain size, which slid from the beginning of the text to the end, creating a map of features. The authors [23] introduced a method of learning from a small sample to classify the text. The authors [24, 25] developed a model of textual steganography by combining text with concealment of information and achieved favorable results. The authors [26] used the temporal functions of several objects based on LSTM to detect spam. The results obtained in [26] showed efficiency in the tonality analysis in long texts.

Parallel DenseNet is proposed in [27] based on traditional tightly coupled convolutional networks to implement short text tone analysis. In particular, this paper proposes two new feature extraction units based on DenseNet and a multiscale convolutional neural network. This model is able to extract both local and global short text features by combining output and features extracted using the parallel feature extraction block and then sending the combined features to the final classifier.

The principle of text tonality analysis based on the bag-of-words model is to put all the words in one package, the so-called word bag. When a word appears in a sentence, the position of that word in the vector is 1 and the position of the other words is 0. In this case, the words in the sentence are out of order. Therefore, the bag-of-words model was developed in methods of distinguishing features, such as tagging part of speech (POS) and tagging phrases N-grams. Tagging a part of speech, also known as grammatical tagging, is the process of marking words in a text (corpus) as corresponding to certain parts. N-gram tagging of phrases is based on the fact that one word depends on several other words. When denoting a word, this word is usually combined with the previous word.

GloVe technology [28] allows to obtain the corresponding vector of fixed length for each word in the text data using statistical information about the word in the data.

Let the size of the dictionary be equal to V . All words found in the data are numbered $\overline{1, V}$. A word-word co-occurrence matrix is formed $X \in R^{V \times V}$, where x_{ij} – indicates how many times a word i is used in the context of the word j . Word a occurs in the context of word b , if there is a part of the text with no more than nine words between them.

Let's mark $X_i = \sum_{k=1} x_{ik}$ (sum of the row i). Then the probability that word j occurs in the context of word i is $P_{ij} = P(j|i) = \frac{x_{ij}}{X_i}$.

It should be noted that if the word occurs in the context of the word k more often than a word j occurs in the context of the word k , so $\frac{P_{ik}}{P_{jk}} > 1$, $\frac{P_{jk}}{P_{ik}} < 1$.

Let's build a function $F(w_i, w_j, \hat{w}_k)$ that shows which of the words i or j more likely to occur in the context of the word k . w_i, w_j, \hat{w}_k – vector representation of words i, j and k .

$$F(w_i, w_j, \hat{w}_k) = \frac{P_{ik}}{P_{jk}}. \quad (7)$$

The authors of the GloVe model suggested using $F((w_i - w_j)^T \hat{w}_k) = \frac{F(w_i^T \hat{w}_k)}{F(w_j^T \hat{w}_k)}$,
 $F(w_i^T \hat{w}_k) = P_{ik} = \frac{x_{ik}}{X_i}$.

Then you can choose $F(x) = \exp(x)$ as a function $F(\cdot)$ and choose the vector w_i that $w_i^T \hat{w}_k = \log(P_{ik}) = \log(x_{ik}) - \log(X_i)$.

Now, given that $\log(X_i)$ is fixed, we rewrite the problem as follows $w_i^T \hat{w}_k + b_i + \hat{b}_k = \log(x_{ik})$, $b_i + \hat{b}_k = \log(X_i)$.

As a result, the authors use the loss function J and adjust the model using an algorithm AdaGrad [4].

Function $f(x)$ must meet the following requirements: $f(0)=0$; $f(x)$ – does not decrease; $f(x)$ – relatively small for large values x .

The authors used the following: $f(x) = \begin{cases} \left(\frac{x}{x_{\max}}\right)^\alpha, & x < x_{\max} \\ 1, & x > x_{\max} \end{cases}$.

The parameters were chosen empirically: $\alpha = \frac{3}{4}$, $x_{\max} = 100$.

3. Problem statement

The problem of classification of textual information is formulated as follows: let there be a finite number of classes in a category $\bar{C} = \{c_1, c_2, \dots, c_m\}$ and a finite set of documents $\bar{D} = \{d_1, d_2, \dots, d_m\}$ and unknown target function f which determines the correspondence for each pair (document, class) $f: \bar{D} \times \bar{C} \rightarrow \{0,1\}$. The task is to find the function f_0 which is as close as possible to the objective function f , that is, a minimum rate is provided $\min \|f - f_0\|$ in Euclidean space. Function f_0 is called a classifier.

Texts tonality is understood as the emotional vocabulary and emotional assessment given by the author in relation to the object. The analysis of the text tonality is of great practical importance: quality avaluation of goods and services based on users' feedback of Internet resources; prevention of extremism and terrorism; analysis of stock markets and forecasting the volatility (variability) of financial assets.

The main task of text tonality analysis is to identify ideas in the text and determine their properties. Opinions are of two types: opinion comparison and direct opinion. Direct opinion contains the author's statement about the object. The formal opinion definition is described as a tuple of 4 elements $\bar{K} = \{o(p), e(f), t, h\}$, where $o(p)$ – orientation or polarity assessment of tonality; $e(f)$ – entity- or feature-object of tonality or its properties f ; t – time – end of the oppinion; h – holder – subject of tonality (author).

Text tonality is assessed as neutral, negative or positive.

In general statistics, volatility is an indicator that characterizes the fluctuations of time series or trends in market prices and incomes over time; composing texts with predetermined emotional characteristics.

There are different types of text classification: subjective; objective; multiscale, i.e. classification according to a multilevel scale and classification according to a binary scale. This article uses the Keras machine learning framework and the Python programming language to solve this problem.

4. An architecture model of a branched convolutional neural network with parallel and serial connections

The architecture of a new hybrid convolutional neural network (Fig. 2) consists of a convolutional neural network block with parallel and serial connections between layers and a maximum pulling layer which is obtained from the matrix of the original text $\bar{x}_0 = (x_1, x_2, \dots, x_m)_{m \times d}$, length m ; where $x_i \in R^d$, $i = 1, 2, \dots, m$. You can use different cores to convolve a sentence and to obtain different

features in the proposed convolutional neural network. The mined features are combined with the matrix of maximum pulling obtained from \bar{x}_0 with the help of a convolutional neural network block. These features are classified using the MLP classifier. It should be noted that the proposed new hybrid convolutional neural network differs from the concise neural network DenseNet [27, 29]. In particular, this model has less convergence time and does not require multiple learning iterations (less computational resource) due to the lack of a dense block DenseNet, which is used to classify the texts tonality [30, 31].

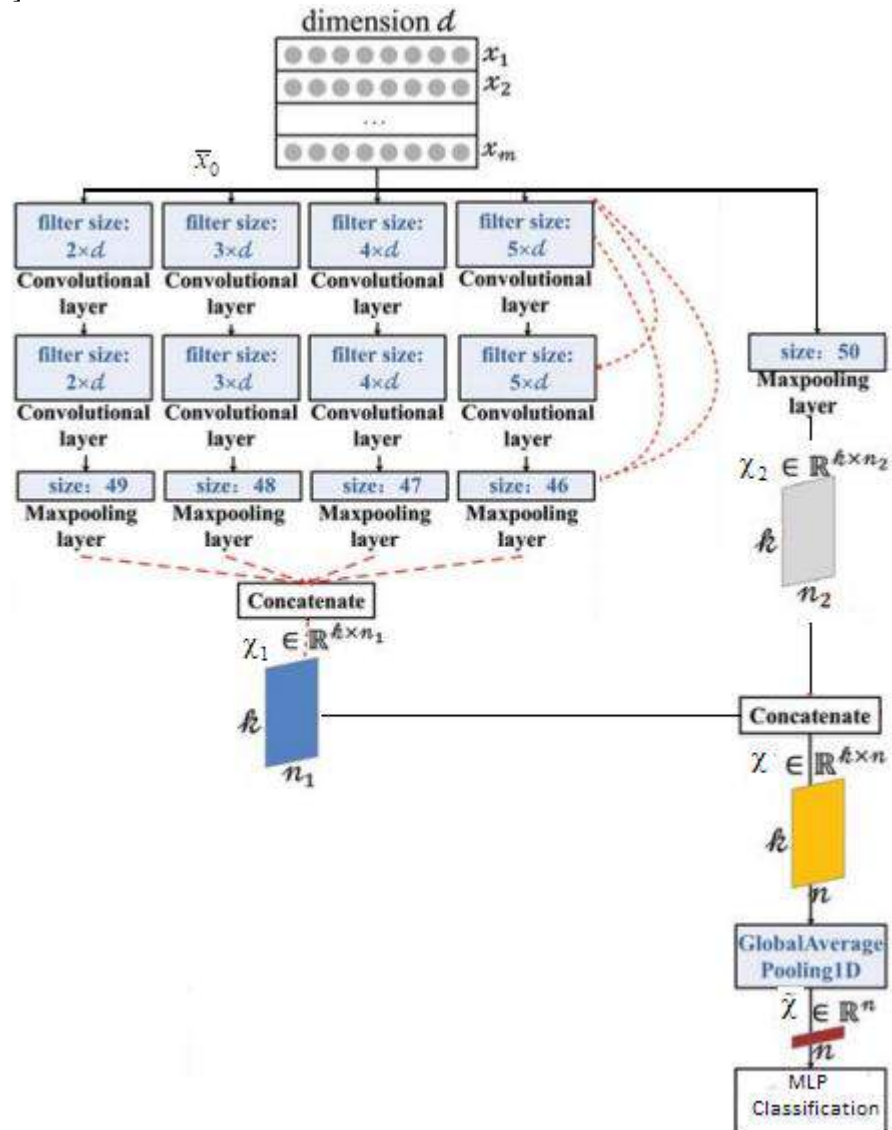


Figure 2: The structure of a hybrid convolutional neural network with parallel and serial connections between layers

The text is entered at the input of the branched convolutional neural network (Fig. 2) $\bar{x}_0 = (x_1, x_2, \dots, x_m)_{m \times d}$, $x_i \in \mathbb{R}^d$, $i = 1, 2, \dots, m$ with length m . This neural network (Fig. 2) consists of two parts: a block of convolutional neural network with parallel and serial connections between layers and a maximum pulling layer. The block of the convolutional neural network consists of layers with different window sizes, which are connected in parallel along the columns and sequentially along the rows of the structure. The input of each convolutional layer in the column consists of the sum of the outputs of all previous layers. A parallel text matrix \bar{x}_0 is applied at the entrance of convoluted layers with dimensions $5 \times d, 4 \times d, 3 \times d, 2 \times d$. To classify the text tonality we will use the average general features, which can be obtained by combining the two characteristics (features obtained from the convolutional network block $\hat{\chi}_1$ and from the maximum pulling layer $\hat{\chi}_2$) due to the global average.

Each convolution subnet is used to mine features using different word combinations, depending on the size of the kernels. In particular, for the kernel $5 \times d$ a combination of 5 words is used to mine the features. Similarly, the input text matrix is presented based on a subnet with kernels $2 \times d, 3 \times d, 4 \times d$.

Input text matrix \bar{x}_0 is introduced into convoluted layers of size $5 \times d, 4 \times d, 3 \times d, 2 \times d$ to mine features.

$$\begin{aligned}\hat{y}_{15} &= f_{5 \times d}(\bar{x}_0) \\ \hat{y}_{14} &= f_{4 \times d}(\bar{x}_0) \\ \hat{y}_{13} &= f_{3 \times d}(\bar{x}_0) \\ \hat{y}_{12} &= f_{2 \times d}(\bar{x}_0)\end{aligned}\quad (8)$$

$\hat{y}_{15}, \hat{y}_{14}, \hat{y}_{13}, \hat{y}_{12}$ – the matrix of features after the first convolutional transformation layer with the sizes of kernels $5 \times d, 4 \times d, 3 \times d, 2 \times d$.

After that, the initial input text matrix is combined with the feature matrices after the convolution transformation, and we obtain new input text matrices $\hat{x}_{15}, \hat{x}_{14}, \hat{x}_{13}, \hat{x}_{12}$

$$\begin{aligned}\hat{x}_{15} &= \text{Cat}([\bar{x}_0, \hat{y}_{15}]) \\ \hat{x}_{14} &= \text{Cat}([\bar{x}_0, \hat{y}_{14}]) \\ \hat{x}_{13} &= \text{Cat}([\bar{x}_0, \hat{y}_{13}]) \\ \hat{x}_{12} &= \text{Cat}([\bar{x}_0, \hat{y}_{12}])\end{aligned}\quad (9)$$

We introduce new input text matrices $\hat{x}_{15}, \hat{x}_{14}, \hat{x}_{13}, \hat{x}_{12}$ in the second convolutional layers with kernels $5 \times d, 4 \times d, 3 \times d, 2 \times d$.

$$\begin{aligned}\hat{y}_{25} &= f_{5 \times d}(\hat{x}_{15}) \\ \hat{y}_{24} &= f_{4 \times d}(\hat{x}_{14}) \\ \hat{y}_{23} &= f_{3 \times d}(\hat{x}_{13}) \\ \hat{y}_{22} &= f_{2 \times d}(\hat{x}_{12})\end{aligned}\quad (10)$$

To obtain new feature matrices, we perform the following operations to combine matrices:

$$\begin{aligned}\hat{x}_{25} &= \text{Cat}([\bar{x}_0, \hat{y}_{15}, \hat{y}_{25}]) \\ \hat{x}_{24} &= \text{Cat}([\bar{x}_0, \hat{y}_{14}, \hat{y}_{24}]) \\ \hat{x}_{23} &= \text{Cat}([\bar{x}_0, \hat{y}_{13}, \hat{y}_{23}]) \\ \hat{x}_{22} &= \text{Cat}([\bar{x}_0, \hat{y}_{12}, \hat{y}_{22}])\end{aligned}\quad (11)$$

After convolution transformations, a pulling operation is performed to obtain new feature matrices.

$$\begin{aligned}\hat{x}^{(1)} &= h_{46}(\hat{x}_{25}) \\ \hat{x}^{(2)} &= h_{47}(\hat{x}_{24}) \\ \hat{x}^{(3)} &= h_{48}(\hat{x}_{23}) \\ \hat{x}^{(4)} &= h_{49}(\hat{x}_{22})\end{aligned}\quad (12)$$

$\hat{x}^{(1)}, \hat{x}^{(2)}, \hat{x}^{(3)}, \hat{x}^{(4)}$ – new matrices of features obtained after the pulling operation $h_{46}, h_{47}, h_{48}, h_{49}$.

After that, the new feature matrices are combined to obtain a matrix of a multiscale block for mining convolutional features.

$$\hat{\chi}_1 = \text{Cat}([\hat{x}^{(1)}, \hat{x}^{(2)}, \hat{x}^{(3)}, \hat{x}^{(4)}]).\quad (13)$$

The *Cat* function describes the merging of matrices $\hat{x}^{(1)}, \hat{x}^{(2)}, \hat{x}^{(3)}, \hat{x}^{(4)}$.

The feature matrix from the maximum pool layer of dimension 50 is described by the formula:

$$\hat{\chi}_2 = h_{50}(\bar{x}_0).\quad (14)$$

We combine matrix features $\hat{\chi}_1$ and $\hat{\chi}_2$ using the function *Cat* to obtain a general feature matrix $\hat{\chi}$

$$\hat{\chi} = \text{Cat}(\hat{\chi}_1, \hat{\chi}_2).\quad (15)$$

And perform a one-dimensional global averaging operation $\hat{\chi}$ to obtain a final feature matrix $\%_0$

$$\%_0 = g(\hat{\chi}). \quad (16)$$

Function g is a one-dimensional average merger. After that, we present the final feature matrix to the neural network classifier (MLP) to classify the text tonality.

5. Computer experiment

Six different data sets were selected for the computer experiment, which are divided into different categories of text tonality. They include:

- GameMultiTweet dataset, this set consists of 12780 parts, which are divided into three categories consisting of 3952 - 915 - 7913 parts.
- SemEval dataset, this set consists of 7967 parts, which are divided into three categories, consisting of 2964 - 1151 - 3852 parts.
- SS-Tweet dataset, this set consists of 4242 parts, which are divided into three categories, consisting of 1953 - 1336 - 953 parts.
- AG News dataset, this set consists of 127,600 parts, which are divided into four categories consisting of 31,900 - 31,900 - 31,900 - 31,900 parts.
- R8 dataset, this set consists of 4203 parts, which are divided into eight categories consisting of 1392 - 241 - 2166 - 20 - 162 - 0 - 72 - 150 parts.
- Yahoo! Answers dataset, this set consists of 350,000 parts, which are divided into ten categories consisting of 23726 - 35447 - 31492 - 35252 - 35546 - 25787 - 25787: 81571 - 23961 - 28706 - 28482 parts.

All of these data sets were randomly divided into three parts: 70% training set, 15% validation set, and 15% testing set. The statistics for each set are given in Table 1.

Table 1
Dataset information

Dataset	Train	Validation	Test	Categories	Avg. length
GameMultiTweet	8964	1917	1917	3	26
SemEval	5577	1195	1195	3	31
SS-Tweet	2870	636	636	3	29
AG News	89320	19140	19140	4	45
R8	2943	630	630	8	66
Yahoo! Answers	245000	52500	52500	10	112

Our model is compared with other models:

- CNN model, consisting of three convolutional layers in which convolutional kernels have the same size.
- TextCNN model proposed in the paper [1].
- FastText model proposed in the paper [17].
- DPCNN model proposed in the paper [21].

In our research, a sentence was converted to a 150x300 matrix using word2vector. Some parameters were set, such as using the adam optimizer, and setting the learning rate to 0.001, dropout rate to 0.2, and L2 loss weight to 10^{-8} . The model batch size was 50 and the number of epochs was 5. If the loss was not reduced in 10 consecutive periods, the training was stopped. In the pre-training word embedding model, 300D word2vector word embedding was used.

Table 2

Comparison of our model with others

Model	GameMultiTweet	SemEval	SS-Tweet	AG News	R8	Yahoo! Answers
CNN	73,5	60,5	50,2	85,6	92,3	47,3
TextCNN	77,5	62,7	51,1	88,9	94,4	49,5
FastText	78,3	63,8	51,4	88,5	96,1	39,8
DPCNN	75,6	47,5	43,2	87,1	88,5	47,5
Our Model	78,5	66,0	52,4	89,7	98,1	51,6

Findings presented in Table 2 show that our model has achieved higher accuracy than its counterparts.

6. Conclusions

A hybrid convolutional neural network for text tonality analysis has been developed. It consists of a convolutional block of parallel and serial connections between layers and a layer of maximum pulling obtained from the matrix of the original text.

It is shown that such a hybrid convolutional neural network mines text features using a convolutional block. Then it mines and classifies the features by combining these features with the original textual features.

It was found that the model of hybrid convolutional neural network has less convergence time and computational resource compared to the parallel DenseNet.

It was proved that a hybrid convolutional neural network with parallel and serial connections between layers provides higher efficiency of text tonality classification in 6 different databases GameMultiTweet, SemEval, SS-Tweet, AG News, R8, Yahoo! Answers compared to other base models.

7. References

- [1] Kim Y. Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1746–1751. URL: <https://arxiv.org/abs/1408.5882>
- [2] Daniel Jurafsky, James H. Martin. Speech and Language Processing, 2021. URL: <https://web.stanford.edu/~jurafsky/slp3>
- [3] P. Liu, X. Qiu, X. Huang. Recurrent neural network for text classification with multi-task learning. In Proc. IJCAI, New York, USA, 2016, pp. 2873–2879. URL: <https://arxiv.org/abs/1605.05101>
- [4] L. Yao, C. Mao, Y. Luo. Graph convolutional networks for text classification. In Proc. AAAI, Hawaii, USA, 2019, pp. 7370–7377. URL: <https://arxiv.org/abs/1809.05679>
- [5] Zulkarnain, Tsarina Dwi Putri. Intelligent transportation systems (ITS): A systematic review using a Natural Language Processing (NLP) approach. Heliyon, 2021, Vol. 7, e08615. doi: 10.1016/j.heliyon.2021.e08615
- [6] J. M. Chenlo, D. E. Losada. An empirical study of sentence features for subjectivity and polarity classification. Information Sciences, 2014, Vol. 280, pp. 275–288. DOI:10.1016/j.ins.2014.05.009
- [7] C. Priyanka, D. Gupta. Identifying the best feature combination for sentiment analysis of customer reviews. In Proc. ICACCI, Mysore, India, 2013, pp. 102–108. DOI:10.1109/ICACCI.2013.6637154
- [8] E. Kouloumpis, T. Wilson, J. Moore. Twitter sentiment analysis: The good the bad and the omg! In Proc. ICWSM, Barcelona, Spain, 2011, pp. 538–541. URL: <https://ojs.aaai.org/index.php/ICWSM/article/view/14185>

- [9] S. Sun, H. Liu, A. Abraham. Twitter part-of-speech tagging using preclassification Hidden Markov model. In Proc. IEEE SMC, Seoul, South Korea, 2021, pp. 1118–1123. DOI:10.1109/ICSMC.2012.6377881
- [10] C. dos Santos, M. Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of COLING. Dublin, Ireland, 2014. pp. 69–78. URL: <https://aclanthology.org/C14-1008>
- [11] Yanyan W., Qun C., Jiquan S., Boyi H., Murtadha A., Zhanhuai Li G. Machine Learning for Aspect-level Sentiment Analysis, 2019. URL: <https://arxiv.org/abs/1906.02502>
- [12] D. Tang, F. Wei, B. Qin, L. Dong, T. Liu et al. A joint segmentation and classification framework for sentiment analysis. In Proc. EMNLP, Doha, Qatar, 2014, pp. 477–487. DOI:10.3115/v1/D14-1054
- [13] F. H. Khan, S. Bashir and U. Qamar. TOM: Twitter opinion mining framework using hybrid classification scheme. Decision Support Systems, 2014, Vol. 57, pp. 245–257. DOI:10.1016/j.dss.2013.09.004
- [14] W. Chamlerwat, P. Bhattarakosol, T. Rungkasiri, C. Haruechaiyasak. Discovering consumer insight from twitter via sentiment analysis. Journal of Universal Computer Science, 2012, Vol. 18, pp. 973–992. URL: <https://www.semanticscholar.org/paper/Discovering-Consumer-Insight-from-Twitter-via-Chamlertwat-Bhattarakosol/b32c462e6a5821c62c852bb42a8730eff880f8cd>
- [15] Yulia Tsvetkova, Manaal Faruqui, Wang Ling, Guillaume Lample, Chris Dyer. Evaluation of Word Vector Representations by Subspace Alignment. Language Technologies Institute Carnegie Mellon University. Pittsburgh, PA, USA, 2021. URL: <https://aclanthology.org/D15-1243.pdf>
- [16] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space, 2014. URL: <https://arxiv.org/abs/1301.3781>
- [17] A. Joulin, É. Grave, P. Bojanowski and T. Mikolov. Bag of tricks for efficient text classification. in Proc. EACL, Valencia, Spain, 2017, pp. 427–431. URL: <https://arxiv.org/abs/1607.01759>
- [18] S. Kombrink, T. Mikolov, M. Karafit and L. Burget. Recurrent neural network based language modeling in meeting recognition. In Proc. INTERSPEECH, Florence, Italy, 2011, pp. 2877–2880. URL: <https://www.semanticscholar.org/paper/Recurrent-Neural-Network-Based-Language-Modeling-in-Kombrink-Mikolov/b4fc91e543ec868658cde6170f1e59c33292e595>
- [19] J. Cheng, X. Zhang, P. Li, S. Zhang, Z. Ding et al. Exploring sentiment parsing of microblogging texts for opinion polling on chinese public figures. Applied Intelligence, 2016, Vol. 45, pp. 429–442. DOI:10.1007/s10489-016-0768-0
- [20] M. Sundermeyer, R. Schlter, H. Ney. LSTM neural networks for language modeling. In Proc. INTERSPEECH, Portland, USA, 2012, pp. 194–197. DOI:10.21437/Interspeech.2012-65
- [21] R. Johnson, T. Zhang. Deep pyramid convolutional neural networks for text categorization. In Proc. ACL, Vancouver, Canada, 2017, pp. 562–570. DOI:10.18653/v1/P17-1052
- [22] S. Wang, M. Huang, Z. Deng. Densely connected CNN with multi-scale feature attention for text classification. In Proc. IJCAI, Stockholm, Sweden, 2018, pp. 4468–4474. URL: <https://www.semanticscholar.org/paper/Densely-Connected-CNN-with-Multi-scale-Feature-for-Wang-Huang/35f0b854901dc6c5a69b271637d302f7db49b79a>
- [23] L. Yan, Y. H. Zheng, J. Cao. Few-shot learning for short text classification. Multimedia Tools and Applications, 2018, Vol. 77, pp. 29799–29810. DOI:10.1007/s11042-018-5772-4
- [24] L. Xiang, S. Yang, Y. Liu, Q. Li, C. Zhu. Novel linguistic steganography based on character-level text generation. Mathematics, 2020, Vol. 8, pp. 1558. DOI:10.3390/math8091558
- [25] Z. Yang, S. Zhang, Y. Hu, Z. Hu, Y. Huang. VAE-Stega: Linguistic steganography based on variational auto-encoder. IEEE Transactions on Information Forensics and Security, 2021, Vol. 16, pp. 880–895. DOI:10.1109/TIFS.2020.3023279
- [26] L. Xiang, G. Guo, Q. Li, C. Zhu, J. Chen et al. Spam detection in reviews using lstm-based multientity temporal features. Intelligent Automation & Soft Computing, 2020, Vol. 26, pp. 1375–1390. DOI:10.32604/iasc.2020.013382
- [27] Luqi Yan, JinHan, Yishi Yue, Liu Zhang, Yannan Qian. Sentiment Analysis of Short Texts Based on Parallel DenseNet. Computers, Materials & Continua, 2021, Vol. 69, pp. 51–65. DOI:10.32604/cmc.2021.016920

- [28] Pennington J. Glove. Global vectors for word representation. EMNLP, 2014, pp. 1532–1543. URL: <https://nlp.stanford.edu/pubs/glove.pdf>.
- [29] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger. Densely connected convolutional networks. in Proc. CVPR, Hawaii, USA, 2017, pp. 4700–4708. URL: <https://arxiv.org/abs/1608.06993>
- [30] Vasyliuk A., Basyuk T. Construction features of the industrial environment control system. Proceedings of the 5th International conference on computational linguistics and intelligent systems (COLINS 2021), Lviv, Ukraine, 2021, Vol. 2870, pp. 1011–1025. URL: <http://ceur-ws.org/Vol-2870/paper76.pdf>
- [31] Basyuk T., Vasyliuk A. Approach to a subject area ontology visualization system creating. Proceedings of the 5th International conference on computational linguistics and intelligent systems (COLINS 2021), Lviv, Ukraine, 2021, Vol. 2870, pp. 528–540. URL: <http://ceur-ws.org/Vol-2870/paper39.pdf>