Investigation of the Deep Learning Approaches to Classify Emotions in Texts

Dmytro Nazarenko[a], Iryna Afanasieva[a], Nataliia Golian[a] and Vira Golian[a]

[a] *Kharkiv National University of Radio Electronics, Nauky Ave. 14, Kharkiv, 61166, Ukraine*

Abstract

The article is devoted to the study of the most popular uses of deep learning - natural language processing, in particular, the extraction of emotions from the text. We compared several types of embeddings and different neural network architectures in solving the problem of classifying emotions from the text. For this, various corpora of text data were collected; they contain markup for the emotional components. Various approaches were analyzed such as Word Embeddings, Bidirectional LSTM, Bidirectional Gated Recurrent Unit, Convolution Neural Network. For comparative analysis, the models were trained and tested on the collected datasets.

As a result, we were found approaches for neural networks creation that provide better results on the test samples.

Keywords [1]

Classification, Data, Emotions, Neural Networks, Deep Learning, Text, Sentiments.

1. Introduction

Emotion detection and text recognition is a mainstream field. Through language processing (NLP), valuable research contributions can be made. Today, documents / data take different forms such as social media posts, news articles, user feedback, and so on, and the content of short texts can be a useful resource for text mining to uncover a variety of issues that can be done with text, including emotions.

Emotion detection and recognition from texts is closely related to sentiment analysis, which are the new areas of study. Sentiment Analysis is a technique for detecting positive, neutral, and negative emotions in texts, while Emotion Analysis looks more deeply and tries to recognize specific type of emotions from anger to happiness, from fear to joy.

The role of recognition in sentiment analysis can be accomplished using lexicon-based approaches, concept-level, or a machine learning approach [1]. In the article we are looking at how we can accomplish this challenge using deep learning methods and a machine learning approach.

Recently, more and more neural network architectures are being created that can use both semantic/syntactic and emotional information by accepting pre-trained word embeddings in order to achieve more efficient results.

In this article, a comparative analysis of state-of-the-art architectures of neural networks has been carried out, as a result of which the efficiency of working with text data for extracting the emotional component has been determined. Using real-world datasets, we performed a thorough evaluation of the models under consideration. Experiments have identified architectures that show the best ability to recognize emotions in text.

COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems, April 22–23, 2021, Kharkiv, Ukraine

EMAIL: dmytro.nazarenko@nure.ua (D. Nazarenko); iryna.afanasieva@nure.ua (I. Afanasieva); nataliia.golian@nure.ua (N. Golian); vira.golan@nure.ua (V. Golian)
ORCID: 0000-0003-4061-0332 (I. Afanasieva); 0000-0002-1390-3116 (N. Golian); 0001-0001-5981-4760 (V. Golian)

2. Related works

Deep learning is a set of machine learning methods (with a teacher, with partial involvement of a teacher, without a teacher, with reinforcement) based on feature / representation learning, rather than specialized algorithms for specific tasks. In the context of textual data, Long short-term memory is the most commonly used deep learning model (LSTM). It is a form of recurrent neural network (RNN) that can handle long-term dependencies. LSTM solves the problem of vanishing or bursting gradients that plagues RNNs. The emotion dataset is first subjected to text preprocessing. Tokenization, stop word deletion, and lemmatization are examples of preprocessing steps. After that, one or more LSTM layers are fed from the embedding layer and then classification is done.

The few attempts to apply deep learning to emotion recognition are considered in the following section. We have researched some implemented deep learning approaches to decide which approaches are state-of-the-art. A robust representation of a tweet was proposed by Meisheri and Dey [3]. Using various pretrained embeddings, two parallel architectures were built to produce the representation. The embedding matrix was generated using emoji2vec [4], GloVe [5], and Character-level embeddings in the first architecture. A BiLSTM [6] was used to feed the resulting matrix. The embedding matrix was developed by the second architecture, which used pretrained GloVe and it was fed into one more BiLSTM, and the results of the BiLSTM was max-pooled. In the SemEval-2018 competition, their model came in second position among the teams.

The next approach was proposed by Wang et al. [7] The approach used a convolutional neural network (CNN) for extracting emotions from texts. The tests used the Chinese blog dataset Ren CECps [8]. The results showed that the CNN could be used for solving emotional recognition tasks and achieves excellent efficiency with the aid of word embedding.

The problem of emotions detection was formulated as a binary classification by Seyeditabari et al. [9]. ConceptNet Numberbatch and fastText were used as word embedding models. A BiGRU layer receives the embedding layer. After pooling operations the results are passed into the dense layers, and the classification is done by a sigmoid layer. The findings show that deep learning models can be used for discovering more in-depth features, leading to substantial improvements in efficiency.

A deep learning model for multi-label emotion detection in microblogs was proposed by Rathnayaka et al. [10]. For preprocessing, they used the ekphrasis method. GloVe, a pre-trained word embedding, was used. Two BiGRU layers are fed from the embedding layer. After outputs concatenation final results of classification are done.

A bidirectional long short-term memory is a model which has two LSTMs, first LSTM takes input in one direction and the second vise versa. BiLSTMs significantly increase the number of data available for the neural network, making sense of the learning algorithm. For example, to know what words are followed and precede a word in a sentence.

CNNs (Convolutional Neural Network) were first developed in the field of neural network image processing, where they achieved ground-breaking results in recognizing objects from a pre-defined category (e.g., cat, bicycle, etc.). Convolution and pooling - two operations that can be presented as feature extractors in a Convolutional Neural Network. We have a 1-dimensional array that represents the text in the case of NLP tasks. The ConvNets' architecture is modified to 1D convolutional-and-pooling operations in this case.

A recurrent neural network (RNN) with a gated recurrent unit (GRU) has some congeniality to an LSTM except its architecture consists of update and reset gates. GRUs are more simple and quicker to train unlike LSTMs because they have fewer parameters. A Bidirectional GRU (BiGRU) - a sequence processing model compound of two GRUs. One takes feedback in a forward direction, while the other takes it backwards. In this BiRNN only the input and forget gates are presented.

As was mentioned above, deep learning based methods mainly use word vectors. Word embeddings are a type of word representation that links a machine's understanding of language to a human's. They've learned text representations in an n-dimensional space, where words with similar meanings are represented similarly. In other words, two related words are represented by nearly identical vectors that are positioned very close together in a vector space. Word2Vec [11], GloVe [12], and FastText [13] are the most widely used methods. Word2Vec is one of the first computationally powerful models for studying trillion-word representations. It outperformed the different n-gram models significantly [14]. GloVe, - global vectors for word representations method, was published later. This was better than Word2Vec because it learned the global hit count instead of the different local context windows that Word2Vec learned. FastText was recently developed to identify and study word representation. Words are treated as the smallest elementary units in Word2Vec and Glove. FastText takes a different approach, treating each word as a set of n-gram characters, also it can handle uncommon terms not found in dictionaries. The analysis of sentiment word embedding [15] is based on a much smaller corpus of textual data. This embedding is used in tasks related to emotion recognition and could find more emotion-related connections between words.

Based on the analysis, it was decided to use 3 main architecture approaches in neural network modeling: BiGRU, BiLSTM and CNN. To work with text data, it was decided to use embeddings in the analyzed models: Word2Vec, GloVe, FastText and Sentiment word embedding.

3. Methods

We used the F-measure metric - the average harmonic of precision and recall. Precision is the proportion of correctly predicted instances among all found, and recall – the proportion of correctly predicted instances relative to the total number of relevants. Also accuracy metric was used to evaluate models performance.

Also weighted modification of F1-score was used - it calculates the F1 score for each class independently but when it adds them together uses a weight that depends on the number of true labels of each class:

$$F1_{class1} * W_1 + F1_{class2} * W_2 + \cdots + F1_{classN} * W_N, \tag{1}$$

To evaluate performance of different architectures we used emotion-annotated datasets [16] from a variety of domains (dialogues, tweets, blogs, and questionnaires):

- Emoint dataset - The dataset that contains texts from twitter posts. The tweets are labeled by crowdsourcing that shows a concentration of anger, joy, sadness, fear [17]. Contains seven thousands of tweets.
- ISEAR dataset – The dataset built on the base of different questionnaires [18]. It has seven classes - joy, fear, anger, sadness, disgust, shame, guilt and contains six thousands of samples.
- StackOverflow dataset - The dataset built on StackOverflow questions and answers, was collected by crowdsourcing. Contains 24K posts and 6 emotions like love, fear, joy, sadness, anger, surprise.
- CrowdFlower dataset - The Twitter dataset published by CrowdFlower. Contains 30K tweets, 5 emotions (neutral, happy, sad, anger, fear). The tweets are annotated with crowdsourcing [19].
- DailyDialogs dataset - The dataset that contains dialogues, labelled with emotions [20]. Contains 13 thousands of dialogues, 100 thousands of speeches, seven emotions, as 0: no emotion, 1: anger, 2: disgust, 3: fear, 4: happiness, 5: sadness, 6: surprise.

Emoint dataset contains over seven thousands of samples. The dataset is compiled and marked up using crowdsource and is in the public domain. Each sample was represented by a text message from Twitter, a tagged emotion and an indicator of the intensity of that emotion.

The intensity indicator shows how much emotion is expressed in a given sample, its value ranges from 0 to 1, where 0 - the complete absence of emotion in the sample, 1 - the text entirely consists of the marked emotion.

When studying the dataset in detail, it was decided to use samples with an emotion intensity over 0.6, so with an intensity less than 0.6, emotion is poorly represented or absent. The cleaned EmoInt dataset for the experiment contains 2245 samples with markup for 4 emotions.

ISEAR dataset contains 7666 samples, each sample is a text of answers collected from various questionnaires and the emotion presented in this text. The entire dataset is used for the experiment.

The StackOverflow dataset contains 24,000 StackOverflow posts, labeled with 6 emotions (love, fear, joy, sadness, anger, surprise) by three raters. Upon detailed study, it was revealed that most posts do not have clear markup - markup is either absent or the label is selected by only one rater. After cleaning, the dataset contains 2929 samples, which are used in further experiment.

CrowdFlower dataset contains 47288 samples, which consist of twitter post text and markup for 5 emotions (neutral, happy, sad, anger, fear). The dataset, cleared of unmarked samples, contains 47206 samples.

DailyDialogs dataset contains 13118 dialogs collected and marked up by crowdsourcing. Each message in the dialogs is marked with 7 emotions (0: no emotion, 1: anger, 2: disgust, 3: fear, 4: happiness, 5: sadness, 6: surprise).

To prepare for the experiment, the dialogues were divided into messages with the corresponding emotional labels - a total of 102968 marked messages.

When studying the data obtained, it was revealed that only 17407 messages have an emotional label, all other messages were marked without emotions.

For the experiment, only messages with emotional labels were used, as shown in table 1.

Table 1
Labels in datasets

	EmoInt	ISEAR	StackOverflow	CrowdFlower	DailyDialogs
anger	474	1096	882	1109	1120
happy	-	-	-	16297	12885
joy	562	1094	491	-	-
love	-	-	1220	-	-
sadness	513	1096	230	15938	1150
surprise	-	-	-	-	1823
fear	696	1095	106	-	174
guilt	-	1093	-	-	-
shame	-	1096	-	-	-
hate	-	-	-	4301	-
neutral	-	-	-	9643	-
disgust	-	-	-	-	353
Total	2245	7666	2929	47206	17407

The datasets were pre-processed where we used normalization and lemmatization [21]. The texts were cleaned of punctuation marks and all letters were switched to lowercase during normalization applying. Then lemmatization was performed - the words were reduced to their normal form, and the stop words were deleted. The next step of data preparation for training was to bring data samples to a form that would be convenient for using them as input parameters of a neural network. The text was vectorized applying the Keras library [22]. Each word in the text was associated with a numeric index in the dictionary and all data samples were presented as numbers of vectors. Each vector was supplemented with zeros to a constant length that the length of the text does not affect the ultimate ability of the neural network to generalize.

4. Embeddings

The experiment involved several types of embeddings: GloVe, Word2Vec, FastText, and Sentiment embedding. For GloVe, Word2Vec, FastText, pre-trained word embedding was used:

- Pre-trained GloVe embedding has 42B tokens, 1.9M vocabulary of unique tokens, uncased and vector representation of each token - 300-dimension English word vectors.
- For Word2Vec approach we used pre-trained Google News corpus with 3 billion running words, word vector model has 3 million 300-dimension English word vectors.
- For FastText approach, we used pre-trained word vectors, which were trained using Common Crawl and Wikipedia. These models were trained using continuous bag of words with position-weights, in 300-dimension English word vectors, with character n-grams of length 5, a window of size 5, and 10 negatives.

Sentiment word embedding was obtained as a result of training the embedding layer of a neural network. For this, a neural network was created, consisting of an embedding layer and a dense layer, as shown in figure 1. The embedding layer is initialized with a zero matrix with the learning function enabled.

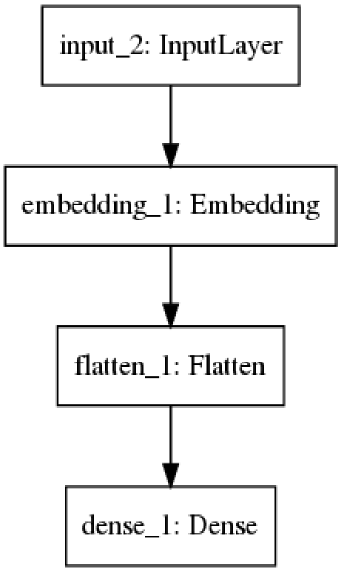

Figure 1: Sentiment word embedding

For training, a universal train sample was created, which is a composition of all datasets, so that the data does not intersect with test samples in the future. The resulting layer is a Dense layer with activation function Softmax, the number of neurons is equal to the total number of classes in all samples - 12. We used the backpropagation algorithm for training and as the loss function we use categorical cross-entropy.

Hyperparameters were tuned for better performance of the trained embedding.

Table 2
Hyperparameter tuning

Hyperparameter	Value 1	Value 2	Value 3
Optimizer	**Adam**	Adadelta	RMSprop
Batch size	50	**100**	300
Learning rate	**0.001**	0.1	10
Epochs	15	10	**5**
Loss function	Categorical crossentropy	-	-
Last layer activation	Softmax	-	-

We determined which hyperparameter was better by training and testing the same model with different values of the corresponding hyperparameter. For testing we used an F1-score metric. As a result, we chose a model trained with Adam optimizer (F1-score: 0.57, Adadelta - 0.53, RMSprop - 0.54) with batch size equaled 100, with learning rate 0.001 and trained for 5 epochs.

5. Train & Test

For a comparative analysis, a number of experiments were carried out on various models to solve the problem of classifying emotions based on text data:

- BiLSTM
- CNN
- BiGRU

The dropout layer was included in front of the output layer that helps to improve the learning quality of the models [23]. The main goal of the dropout method is to prevent overfitting and it is usually used for the regularization of artificial neural networks. The core of the method is that during the learning from the neural network subnet is randomly distinguish and training for the subnet is afforded. The training subnet provided from excluding of neurons from the initial network with probability p, and the probability that the neuron will be in the network equal to 1-p and the excluded neurons do not contribute to any stages of the backpropagation algorithm. So excluding at least one of the neurons means learning a new neural network. We decided to make the output layer as a full layer with N neurons that equal the labels of classified emotions.

The next one, as the neuron activation function was chosen the Softmax function [24]. Categorical cross-entropy is applied to categorize one label that equals one category for each data point is used. So, a data sample can be related to one class exclusively [25]. Categorical cross-entropy is used with the Softmax function. During the comparison, we used the same hyperparameters settings in default for all models, as shown in table 3.

Table 3
Settings of hyperparameters

Hyperparameter	Value

Optimizer	Adam
Batch size	150
Learning rate	0.001
Validation split	0.2
Loss function	Categorical crossentropy

As optimizer was decided to use Adam optimization algorithm. All datasets were splitted into train/test datasets in proportion 80/20, also for validation of the train steps was used train/validation split in the same proportion.

5.1. BiLSTM

We created the neural network architecture, as shown in figure 2. The layers are embedding, bidirectional long short-term memory (bidirectional LSTM), pooling, dropout layers, two dense layers, and the next way is to depict this layers.

Firstly input is going through the embedding layer. To compare work of this architecture with different embedding layer we train models for each type of pretrained embedding: GloVe, Word2Vec, FastText, EWE.

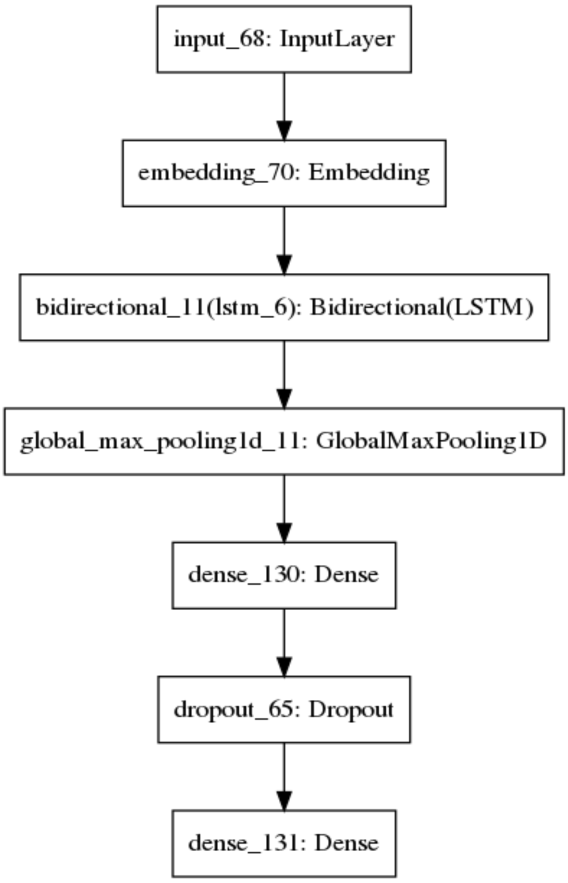

Figure 2: Architecture of BiLSTM model

The next layer is BiLSTM. We used the most popular function - the max function in GlobalMaxPooling [26].

We train and test model for each type of investigated embedding and for each type of datasets. First, we trained the model with GloVe embedding. For the train and test we frozed its embedding layer to not overfit the generalization capability of the model, as shown in table 4.

Table 4

BiLSTM with Glove results

Dataset	Accuracy	Weighted F1-score
EmoInt	0.85	0.86
ISEAR	0.55	0.56
StackOverflow	0.71	0.76
CrowdFlower	0.62	0.61
DailyDialogs	0.85	0.85

The maximum F1-score is 0.86 for DailyDialogs, the minimum score is 0.56 - for ISEAR. The most balanced distribution of predictions by classes belongs to EmoInt - each of the classes has at least 82% true positive outcomes. The confusion matrix for EmoInt is presented in figure 3.

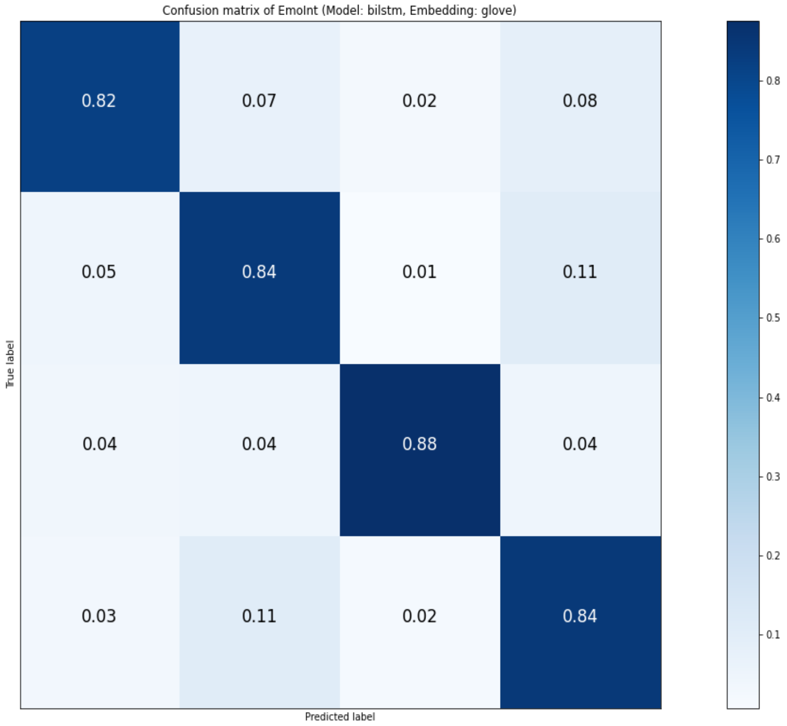

Figure 3: EmoInt confusion matrix

ISEAR predictions have a bigger variance of true positive outcomes ranging from 32% to 86%. In turn, most of the CrowdFlower classes have a close percentage of true positive cases - from 66% to 70%, but `neutral` class has only 44%.

The next testing embedding was sentiment embedding. As we trained it before, we disable training for this layer and combined it with BiLSTM network architecture.The results were shown in table 5.

Table 5

BiLSTM with Sentiment embedding results

Dataset	Accuracy	Weighted F1-score
EmoInt	0.79	0.79
ISEAR	0.53	0.53
StackOverflow	0.70	0.72
CrowdFlower	0.55	0.54
DailyDialogs	0.82	0.83

The most balanced distribution of predictions by classes belongs to EmoInt. The maximum F1-score is 0.83 for DailyDialogs, the minimum score is 0.53 - for ISEAR. Most of true positive outcomes in DailyDialogs predictions belong to the `happy` class - 94% of true positive predictions. EmoInt predictions vary from 71% to 85% accuracy among all classes. Crowdflower and ISEAR classes predictions both vary from 43% to 68%. StackOverflow predicted classes accuracy varies from 40% to 89%.

After sentiment embedding, BiLSTM with FastText embedding was investigated with default hyperparameters as shown in table 6.

Table 6

BiLSTM with FastText embedding results

Dataset	Accuracy	Weighted F1-score
EmoInt	0.42	0.46
ISEAR	0.39	0.40
StackOverflow	0.54	0.64
CrowdFlower	0.61	0.62
DailyDialogs	0.80	0.83

The maximum F1-score is 0.83 for DailyDialogs, the minimum score is 0.40 - for ISEAR. The most balanced distribution of predictions by classes belongs to the CrowdFlower dataset. The accuracy of predicted classes varies from 65% to 69%. For DailyDialogs accuracy distribution among classes is almost similar as in the case of sentiment embedding.

We trained the model with Word2Vec embedding and investigated the performance of it using the test data. The results were shown in table 7.

Table 7

BiLSTM with Word2Vec embedding results

Dataset	Accuracy	Weighted F1-score
EmoInt	0.74	0.75
ISEAR	0.55	0.57
StackOverflow	0.66	0.76
CrowdFlower	0.62	0.63
DailyDialogs	0.83	0.84

The most balanced distribution of predictions by classes belongs to CrowdFlower dataset. The maximum F1-score is 0.84 for DailyDialogs, the minimum score is 0.57 - for ISEAR.

To sum up, we compare performance of the BiLSTM model for each embedding type by weighted F1-score metric. The results were shown in table 8.

Table 8

Generalized BiLSTM results

Dataset	GloVe	FastText	Word2Vec	Sentiment
EmoInt	**0.85**	0.46	0.75	0.79
ISEAR	0.56	0.40	**0.57**	0.53
StackOverflow	**0.76**	0.64	**0.76**	0.72
CrowdFlower	0.61	0.62	**0.63**	0.54
DailyDialogs	**0.86**	0.83	0.84	0.83

The best performance was showed by GloVe and Word2Vec, but Word2Vec showed better performance for more balanced datasets - CrowdFlower and ISEAR (both haven’t got explicit data outliers), which means better generalization capability.

5.2. CNN

CNN model consists of embedding layer и convolution layer, hidden layers, Pooling layer, and dropout. The model utilizes the last hidden state for emotion recognition. The architecture of the network is presented in figure 4.

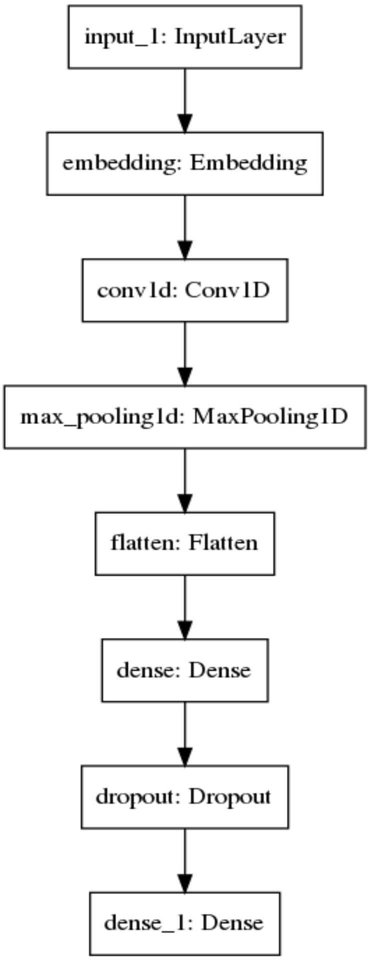

Figure 4: Architecture of CNN model

We train and test model for each type of investigated embedding and for each type of datasets. First we train the model with GloVe embedding. For train and test we froze its embedding layer to do not overfit the generalization capability of the model, as shown in table 9.

Table 9

CNN with GloVe embedding results

Dataset	Accuracy	Weighted F1-score
EmoInt	0.54	0.57
ISEAR	0.33	0.35
StackOverflow	0.59	0.66
CrowdFlower	0.55	0.55
DailyDialogs	0.80	0.83

Maximum F1-score was for DailyDialogs - 0.83, minimum - 0.35 for ISEAR. For Emoint accuracy for classes prediction varies in range between 61% and 88%. Most class predictions of ISEAR dataset have less than 50% accuracy, weighted accuracy among all classes - 33%. StackOverflow predictions has only 2 classes which accuracy are more than 50% - love (73%) and anger (80%), this is due to the fact that these classes are mostly represented in the dataset. CrowdFlower predicted classes accuracy vary from 41% (`neutral`) to 70% (`fear`).

The next embedding that we trained the model with was sentiment embedding, as shown in table 10.

Table 10

CNN with Sentiment embedding results

Dataset	Accuracy	Weighted F1-score
EmoInt	0.78	0.78
ISEAR	0.50	0.50
StackOverflow	0.69	0.71
CrowdFlower	0.54	0.54
DailyDialogs	0.81	0.81

Maximum F1-score was 0.81, minimum - 0.50. For EmoInt all classes have more than 70% accuracy. ISEAR predictions are in range between 37% and 67% for each class. CrowdFlower and DailyDialogs have similar results like GloVe. StackOverflow predictions for `neutral` class have better accuracy than GloVe approach 80% and 73% respectively.

FastText predictions have less weighted F1-score than sentiment embedding approach for all tested datasets except CrowdFlower - neutral class has 38% accuracy while sentiment approach has 30% for the same class and happy class has 70% vs 58% in sentiment approach, as shown at table 11.

Table 11

CNN with FastText embedding results

Dataset	Accuracy	Weighted F1-score
EmoInt	0.41	0.43
ISEAR	0.26	0.27
StackOverflow	0.56	0.64
CrowdFlower	0.57	0.57
DailyDialogs	0.76	0.80

For Word2Vec minimum weighted F1-score was obtained for ISEAR dataset. Its predicted accuracy for each class varies from 18% (`guilt`) to 56% (`disgust`). Emoint prediction accuracies vary from 22% (anger) to 75% (sadness). For other datasets differences are insignificant, as shown at table 12.

Table 12

CNN with Word2Vec embedding results

Dataset	Accuracy	Weighted F1-score
EmoInt	0.48	0.53
ISEAR	0.36	0.36
StackOverflow	0.58	0.64
CrowdFlower	0.56	0.56
DailyDialogs	0.79	0.82

Comparing test results for each type of embedding there was found for CNN architecture Sentiment embedding is better than other embedding methods, because it gives a significant gain for sparsed datasets like EmoInt and StackOverflow and gives better performance for balanced dataset ISEAR. For CrowdFlower and DailyDialogs all embeddings showed similar results, as shown at table 13.

Table 13

Generalized CNN results

Dataset	GloVe	FastText	Word2Vec	Sentiment
EmoInt	0.57	0.43	0.53	**0.78**
ISEAR	0.35	0.27	0.36	**0.50**
StackOverflow	0.66	0.64	0.64	**0.71**
CrowdFlower	0.55	**0.57**	0.56	0.54
DailyDialogs	**0.83**	0.80	0.82	0.81

5.3. BiGRU

The neural network architecture was created as shown in figure 5. Our interconnected layers are consist of embedding, bidirectional GRU, pooling, dropout layers and two dense layers, so let us describe each layer more detail.

Firstly input is going through the embedding layer. To compare work of this architecture with different embedding layer we train models for each type of pretrained embedding: GloVe, Word2Vec, FastText, EWE.

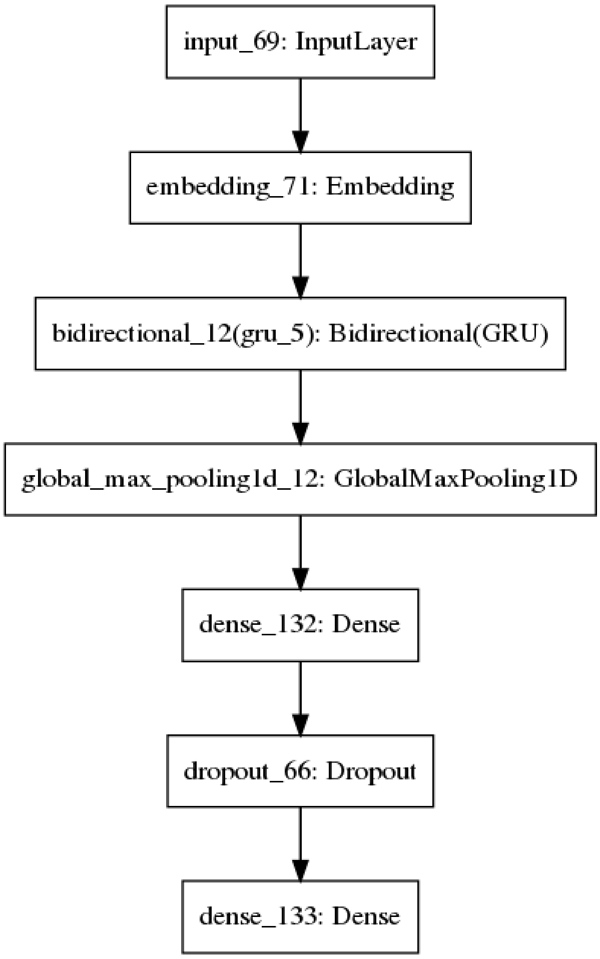

Figure 5: Architecture of BiGRU model

The approach with BiGRU and GloVe has maximum weighted F1-score for EmoInt - 0.90, as shown at table 14. Prediction accuracies vary between 76% and 98%. The minimum F1-score for ISEAR - 0.53: accuracies between 39% and 74%.

Table 14

BiGRU with GloVe embedding results

Dataset	Accuracy	Weighted F1-score
EmoInt	0.90	0.90
ISEAR	0.54	0.53
StackOverflow	0.70	0.78
CrowdFlower	0.61	0.62
DailyDialogs	0.84	0.85

The approach with sentiment embedding shows equal or worse performance for each type of dataset for both averaged accuracy and weighted F1-score metrics. Maximum weighted F1-score is for EmoInt (0.85), minimum - balanced datasets ISEAR (0.53) and CrowdFlower (0.55), as shown at table 15.

Table 15

BiGRU with Sentiment embedding results

Dataset	Accuracy	Weighted F1-score
EmoInt	0.85	0.85
ISEAR	0.54	0.53
StackOverflow	0.70	0.71
CrowdFlower	0.55	0.55
DailyDialogs	0.82	0.83

BiGRU approach with FastText embedding maximum weighted F1-score belongs to DailyDialogs dataset and minimum for both EmoInt and ISEAR datasets we can see at table 16. This approach shows the worst performance for EmoInt dataset - the model ovefits on the train data for the `joy` class and most of unlabeled test data were labeled as `joy`. More detailed information about distribution is shown on the confusion matrix in figure 6.

Table 16

BiGRU with FastText embedding results

Dataset	Accuracy	Weighted F1-score
EmoInt	0.37	0.44
ISEAR	0.43	0.44
StackOverflow	0.62	0.71
CrowdFlower	0.61	0.62
DailyDialogs	0.82	0.85

And as we can see, BiGRU with FastText shows better performance than sentiment approach and similar performance with GloVe approach on big balanced dataset CrowdFlower. But the weighted F1-score is much worse for small datasets EmoInt and ISEAR than corresponding models. The model prediction ability for StackOverflow, that we counted from collected metrics, shows that the model revealed worse results for the dataset than the GloVe. To sum up, BiGRU with FastText works better for bigger datasets.

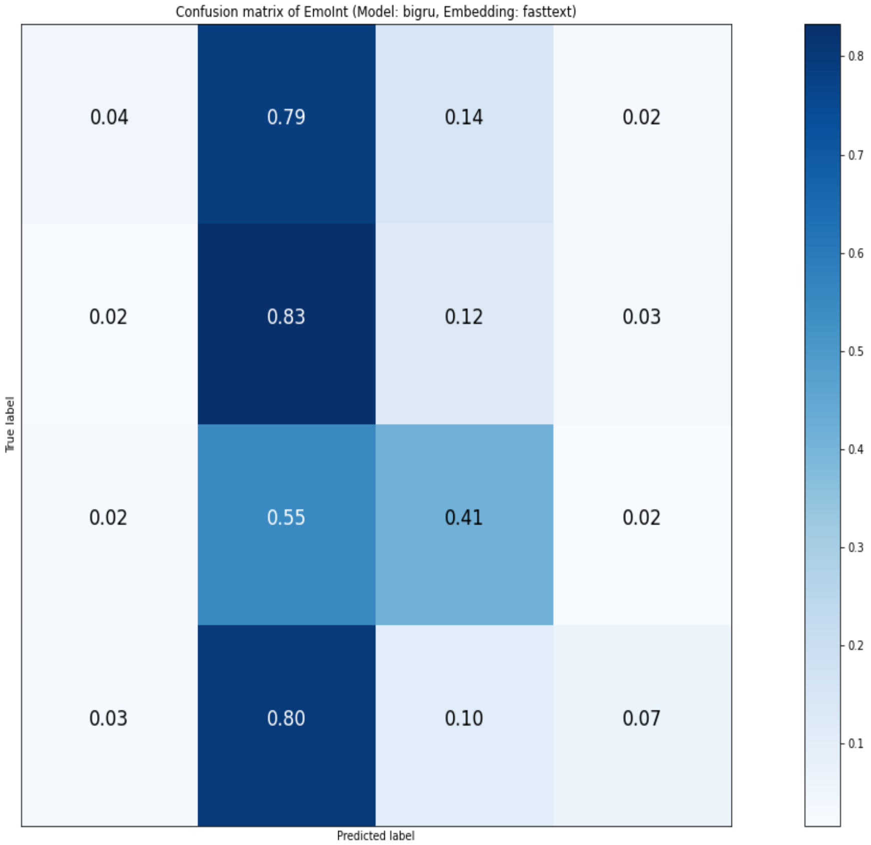

Figure 6: Confusion matrix for EmoInt

For Word2Vec the minimum weighted F1-score belongs to ISEAR dataset - the vary of predicted classes accuracy is between 12% and 54%, as shown at table 17.

Table 17
BiGRU with Word2Vec embedding results

Dataset	Accuracy	Weighted F1-score
EmoInt	0.38	0.51
ISEAR	0.24	0.29
StackOverflow	0.50	0.60
CrowdFlower	0.58	0.60
DailyDialogs	0.79	0.84

After analyzing all possible embedding options and comparing the results, it was concluded that the best combination of approaches for BiGRU is GloVe usage. BiGRU combined with GloVe showed the best performance for each type of datasets, as shown at table 18.

Table 18

Generalized BiGRU results

Dataset	GloVe	FastText	Word2Vec	Sentiment
EmoInt	**0.90**	0.44	0.51	0.85
ISEAR	**0.53**	0.44	0.29	**0.53**
StackOverflow	**0.78**	0.71	0.60	0.71
CrowdFlower	**0.62**	**0.62**	0.60	0.55
DailyDialogs	**0.85**	**0.85**	0.84	0.83

BiGRU with FastText shows good performance for datasets which contains more than 10000 samples - CrowdFlower and DailyDialogs, but for smaller datasets it performed worse than with GloVe embedding.

6. Summary

In order to conduct a final analysis, we collected all the results and compared them with each other for each dataset, as shown at table 19. Datasets: E - EmoInt, I - ISEAR, S - StackOverflow, D - DailyDialogs. We use a weighted F1-score as a model performance representation metric.

Table 19

Generalized results

Dataset	E	I	S	C	D
BiLSTM + GloVe	0.85	0.56	0.76	0.61	**0.86**
BiLSTM + FastText	0.46	0.40	0.64	0.62	0.83
BiLSTM + Word2Vec	0.75	**0.57**	0.76	**0.63**	0.84
BiLSTM + Sentiment	0.79	0.53	0.72	0.54	0.83
CNN + GloVe	0.57	0.35	0.66	0.55	0.83
CNN + FastText	0.43	0.27	0.64	0.57	0.80
CNN + Word2Vec	0.53	0.36	0.64	0.56	0.82
CNN + Sentiment	0.78	0.50	0.71	0.54	0.81
BiGRU + GloVe	**0.90**	0.53	**0.78**	0.62	0.85
BiGRU + FastText	0.44	0.44	0.71	0.62	0.85
BiGRU + Word2Vec	0.51	0.29	0.60	0.60	0.84
BiGRU + Sentiment	0.85	0.53	0.71	0.55	0.83

It was found that the model architecture with BiLSTM and Word2Vec as Embedding performs better for datasets with balanced classes - means that the dataset has relatively enough samples for each class. Also, it was revealed that for datasets with sparse classes such as EmoInt and StackOverflow the best performance was shown by BiGRU with GloVe architecture.

7. Conclusions

In the article we made an investigation of the state-of-the-art deep learning approaches for emotion recognition from textual data. We explore different neural network architectures and evaluate their performance.

For baseline approaches there were investigated the most popular word embeddings and baseline neural network approaches. For embeddings we used 3 pretrained embeddings: GloVe, FastText, Word2Vec and one embedding on the train set of initial data. 3 network approaches were chosen: BiLSTM, BiGRU, CNN.

The performance metrics used here were the rates of recall, precision, weighted accuracy, F1-score (weighted and unweighted) and a confusion matrix. For evaluation models performance we collected 5 datasets with emotion labels: EmoInt, ISEAR, StackOverflow, CrowdFlower and DailyDialogs. Datasets were labeled in summary in 12 emotion classes: anger, happy, joy, love, sadness, surprise, fear, guilt, shame, hate, neutral, disgust. Each of the datasets were splitted in train, validation and test subdatasets.

During the research, 12 neural network architectures were created based on the selected embeddings and deep learning approaches. For each architecture, the procedure for evaluating the quality indicator was as follows: choosing a dataset, training on labeled data, testing on unlabeled data, and collecting metrics. Then all collected metrics for datasets were combined and analyzed.

As a result, it was found that the use of BiLSTM and Word2Vec in the neural network architecture provides better results on balanced dataset. It showed best performance for ISEAR and CrowdFlower datasets - 0.57 and 0.63 weighted F1-score respectively. For sparsed datasets (EmoInt and StackOverflow) it was found that the architecture with the best performance scores is BiGRU with pretrained GloVe embedding - 0.90 and 0.78 F1-score respectively. For DailyDialogs dataset the best performance was shown by BiLSTM model with GloVe embedding, but the result score of BiLSTM + Word2Vec and BiGRU + GloVe models have significantly close results - 0.86, 0.84 and 0.85 weighted F1-score respectively.

8. References

[1] W. Medhat, A. Hassan, H. Korashy, Sentiment Analysis Algorithms and Applications: A Survey, Ain Shams Engineering Journal (2014): 1093-1113.

[2] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT Press, Cambridge, 2016.

[3] H. Meisheri, L. Dey, TCS research at SemEval-2018 task 1: learning robust representations using multi-attention architecture, in: Proceedings of the 12th international workshop on semantic evaluation, Association for Computational Linguistics, 2018, pp. 291–299.

[4] B. Eisner, T. Rocktäschel, I. Augenstein, M. Bosnjak, S. Riedel, emoji2vec: learning emoji representations from their description, in: Proceedings of the fourth international workshop on natural language processing for social media, Association for Computational Linguistics, 2016, pp. 48–54.

[5] J. Pennington, R. Socher, C. Manning, Glove: global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), Association for Computational Linguistics, 2014, pp. 1532–1543.

[6] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional lstm and other neural network architectures, Neural Netw 18(5), 2005, pp. 602–610.

[7] Y. Wang, S. Feng, D. Wang, G. Yu, Y. Zhang, Multi-label chinese microblog emotion classification via convolutional neural network, in: Web technologies and applications, APWeb 2016, vol 9931, Lecture notes in computer science. Springer, Cham, 2016, pp. 567–580.

[8] C. Quan, F. Ren, A blog emotion corpus for emotional expression analysis in Chinese, Comput Speech Lang 24(4), 2010, pp. 726–749.

[9] A. Seyeditabari, N. Tabari, S. Gholizadeh, W. Zadrozny, Emotion detection in text: focusing on latent representation, 2019.

[10] P. Rathnayaka, S. Abeysinghe, C. Samarajeewa, I. Manchanayake, M.J. Walpola, R. Nawaratne, T. Bandaragoda, D. Alahakoon, Gated recurrent neural network approach for multilabel emotion detection in microblogs, 2019.

[11] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013.

[12] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), Doha, Qatar, 2014, pp. 1532–1543.

[13] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5, 2017, pp.135-146.

[14] T. Mikolov, A. Deoras, S. Kombrink, L. Burget, J. Černocký, Empirical evaluation and combination of advanced language modeling techniques, in: Twelfth Annual Conference of the International Speech Communication Association, 2011.

[15] A. Agrawal, A. An, M. Papagelis, Learning emotion-enriched word representations, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 950-961.

[16] L.A.M. Bostan, R. Klinger, An Analysis of Annotated Corpora for Emotion Classification in Text, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 2104-2119.

[17] S.M. Mohammad, F. Bravo-Marquez, Emotion intensities in tweets, 2017.

[18] K.R. Scherer, H.G. Wallbott, Evidence for universality and cultural variation of differential emotion response patterning, Journal of personality and social psychology, 66(2), 1994, p.310.

[19] S. Mohammad, S. Kiritchenko, Understanding emotions: A dataset of tweets to study interactions between affect categories, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018), 2018.

[20] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, S. Niu, Dailydialog: A manually labelled multi-turn dialogue dataset, 2017.

[21] A. Eija, Word Normalization and Decompounding in Mono- and Bilingual IR//Information Retrieval, volume 9, 2006, Issue 3. pp. 249–271.

[22] N. Ketkar, Introduction to Keras, in: Deep Learning with Python, Apress, Berkeley, CA, 2017, pp. 97-111.

[23] G. Hinton, Improving neural networks by preventing co-adaptation of feature detectors, 2012, p. 18.

[24] G. Bolin, P. Lacra, On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning, 2017, p. 10.

[25] K. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012, pp. 57-571.

[26] B. Graham, Fractional Max-Pooling, 2015, p. 10.