Detecting emotion intensity in tweets

Olha Kaminska
dil-delada@ukr.net

Viacheslav Komisarenko
komysarenkov@ukr.net

Abstract

In this work we tried to build models that determine emotions intensity based on texts from the most popular social network - Twitter.

We propose model that predicts for four basic emotions - anger, fear, joy or sadness - their intensity in given short text - tweet.

In the beginning we had prepared training dataset for training and analogical development data for testing. Wide variety of features that fit this task were extracted, most of them based on word embedding. After that, different machine learning models were implemented and choosing the best one.

As a result, working models and good visualization were obtained, report and poster for presentation were prepared.

1 Introduction

Detecting emotions intensity is important and well-known task in natural language processing. Furthermore, this problem is interesting not only for scientists, but also actively used in business.

Social networks are perfect sources for gathering people thoughts. It is modern place for communication where everyone can express opinion about anything. Text is good way for sharing feeling and ideas, but in our days the main tools for these purposes are hashtags, emoticons and even links.

Twitter is one of the most popular social network, millions tweets are written every day, which is a huge field for analysis. Societys reaction to global news can be concluded based on collecting people's opinions.

In this work we propose models that predicts emotions' (anger, fear, joy and sadness) intensity in given tweets. More precisely, tweet and emotion are given as an input, and intensity should be determined in a form of real value between zero and one.

1.1 Related works

(Davidov et al., 2010) used 50 twitter tags and 15 smileys as class labels and build classifier for detecting sentiment of short-text tweets. They used single-word, n-gram, punctuation and pattern features and perform evaluation with the help of human judges.

In (Mohammad and Bravo-Marquez, 2017) authors work only with text data. For each emotion they developed 50 to 100 terms that associate emotion at different intensity level (e.g. angry, mad, frustrated for anger). Also, authors analyzed impact of removal emotion word hashtags.

In (Duppada and Hiray, 2017) authors combines lexical, syntactic and pre-trained word embedding features and used AdaBoost with XGBoost as base regressor. For tuning hyperparameters was used 10-fold cross-validation with optimization Pearsons correlation score.

2 Methodology

Source text with emotional tweet was separated by emotions, preprocessed and tokenized. From obtained tokens were extracted useful features, which then where vectorized. Input emotions' intensity were used as regression labels.

All vectors and labels were used in several machine learning regression models, their quality was evaluated by **RMSE** (Root-Mean-Square Error) parameter:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y_i})^2},$$

where n - number of tweets,
y_i - real intensity,
$\bar{y_i}$ - predicted intensity.

Score function RMSE was chosen for evaluation results, because this is standard technique for measuring quality of regression.

In the end for every emotions was chosen the best model. Obtained results were tested on new data with intensity of all four emotions of those data as a result.

3 Experiments

During the project a lot of analyzing tasks were solved: analysis and preprocessing input data, preparation it for using in the model, extracting large amount of features, creation several different machine learning models and analysis obtained results.

3.1 Data

3.1.1 Data format

At the beginning two folders were given, first with train data and second with validation data. Each folder contains four files in .txt format [1], one per emotion: anger, fear, joy, sadness. Structure of each .txt file is illustrated in Table 1.

ID	Tweet	Emotion	Score
1	My life is cool! <3	joy	0.91
2	I lose my key... :(	sadness	0.73
3	Traffic jams annoing!	anger	0.86
4	I heard some voices!	fear	0.65
...	...	...	...

Table 1: Content of input .txt file

ID is a personal unique id for each tweet - this information is useless for analysis. Column Tweet contains unfiltered text of the tweet with all hashtags, emojis etc. Emotion is the same for every row in one document and its mentioned in the name of the file. Column Score shows how strong this emotion is. Its intensity that distributed from 0 to 1.

On Figure 1 presented distribution of emotion intensity in train data.

3.1.2 Data downloading

Files were written in UTF-8 unicode.

Data were read file by file in separate vocabularies. Only second and fourth columns were taken - Tweet and Score. First column was skipped, because it has unique values for each tweet and has no information for analysis, third - because it has the same value for each row in file.

[1]Near 2,000 tweets per emotion for each 'train' file and near 350 tweets per emotion for each 'development' file.

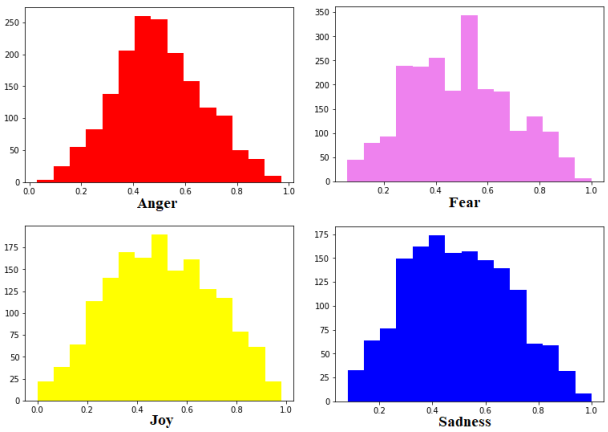

Figure 1: Emotion intensity in train data.

Rows in files were splitted by tabulation. Every score has symbol in the end, so in each line last symbol was skipped and result was saved as float data type.

For each emotion was created one vocabulary in train and development set, in total eight dictionaries. For each vocabulary key was text of tweet, value - score of emotion intensity.

3.1.3 Data preprocessing

Was created function with vocabulary as input and two lists as output: one with tokens, another with score of emotion intensity.

Each key was splitted on tokens and every token has number of checks. Token was deleted if:

- it was one of the standard English stop words (imported from nltk.corpus);
- it was one of the punctuation marks, written in special list by hands;
- it was HTML-tags / mentions of other users / URLs / numbers.

To define which type of token it is, was written mask on the base of regular expressions, with using re library, for performing filtering. It received string of tweet text as input and list of tokens with correct splitting as output.

All emojis and hashtags were left, because they contain emotions and using them as features could improve results.

3.2 Experimental setup

3.2.1 Features selection

From every tweet from each training set were extracted 1803 features. The most of them were based on Word Embedding Features, that was used for every word in each tweet.

For implementation were used the Google News 300-dimensional pre-trained word embeddings. As features were used statistical characteristics of obtained vectors set, listed below.

Extracted features from Word Embeddings:

- Arithmetic mean of the vectors,
- Median of the vectors,
- Minimum for each coordinate among all vectors,
- Maximum for each coordinate among all vectors,
- 25% percentile for each coordinate among all vectors,
- 75% percentile for each coordinate among all vectors,

Each of those features is vector of size 300, this way, in general, were obtained 1800 Word Embedding features.

Also, three more features were selected:

- The number of tokens in the tweet,
- Total number of characters in tweet,
- The average number of symbols in the token.

3.2.2 Models description

After features extraction were created several basic models and their ensembles.

Different regression models were used from 'sklearn' library in Python with special set of parameters for each. Below listed all models with setted parameters.

1. **Support Vector Regression, RBF kernel**
   - Kernel type = 'rbf'[2]
   - Kernel coefficient = 0.1
   - Penalty parameter - grid of values: 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50. [3]
2. **Support Vector Regression, linear kernel**
   - Kernel type = 'linear'
   - Penalty parameter - grid of values: 0.01, 0.1, 1, 10, 100.
3. **Support Vector Regression, polynomial kernel**
   - Kernel type = 'poly'
   - Degree of polynomial function = 2
   - Penalty parameter - grid of values: 0.01, 0.1, 1, 10, 100.
4. **Decision Tree Regressor**
   - The number of features to consider when looking for the best split - grid of values from 10 to 510 with step 100.
5. **K-Neighbors Regressor**
   - The number of neighbors - grid of values from 2 to 20 with step 2.
6. **Random Forest Regressor**
   - The number of trees - grid of values from 10 to 210 with step 50.
7. **Gradient Boosting Regressor**
   - The number of boosting stages to perform - grid of values from 10 to 260 with step 50.

3.3 Results

3.3.1 The best output of models

In this section presented the best results of used models with correspondent parameters.

1. **Support Vector Regression**

   On Figure 2 presented the best results with corresponded Penalty parameter (in the middle of each bar plot) for every type of the kernel (indicated in parentheses after emotions).

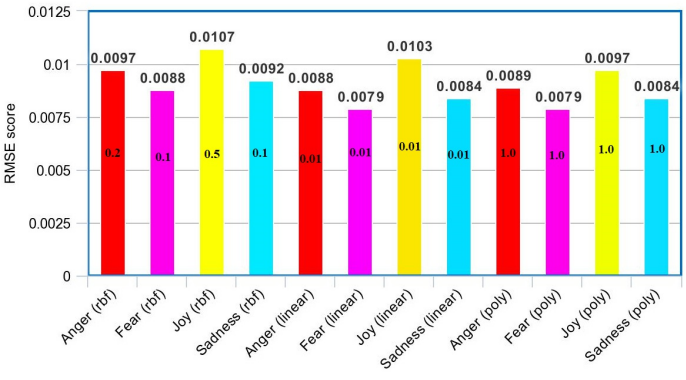

Figure 2: The best results for SVR model.

2. **Decision Tree Regressor**

   On Figure 3 presented the best results with corresponded Number of Features parameter (in the middle of each bar plot).

[2] Radial Basis Function

[3] Based on experiment with this model, we can conclude that there is no significant difference for penalty parameter if we change it linearly. So for other SVR model we changed it exponentially

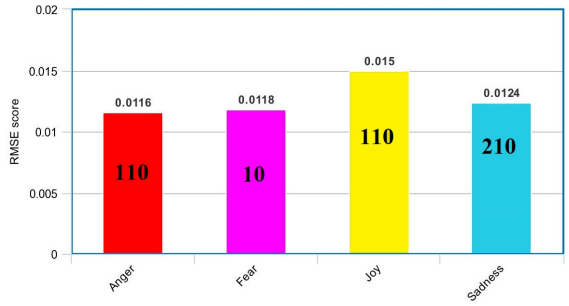

Figure 3: The best results for DT model.

3. **K-Neighbors Regressor**

On Figure 4 presented the best results with corresponded Number of Neighbors parameter (in the middle of each bar plot).

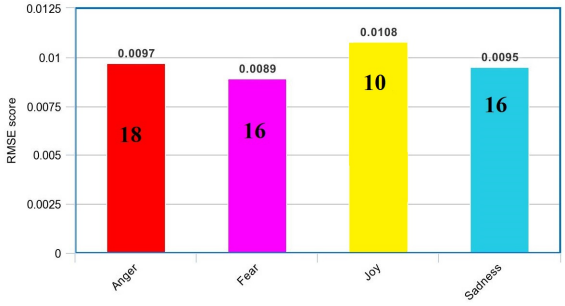

Figure 4: The best results for KNR model.

4. **Random Forest Regressor**

On Figure 5 presented the best results with corresponded Number of Trees parameter (in the middle of each bar plot).

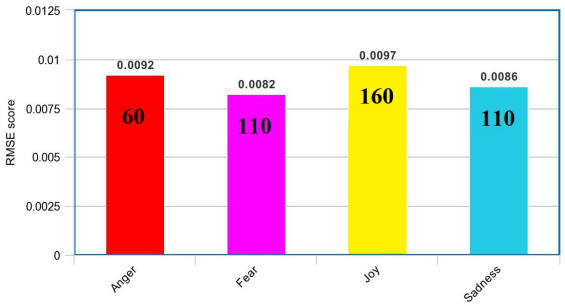

Figure 5: The best results for RFR model.

5. **Gradient Boosting Regressor**

On Figure 6 presented the best results with corresponded Number of Stages parameter (in the middle of each bar plot).

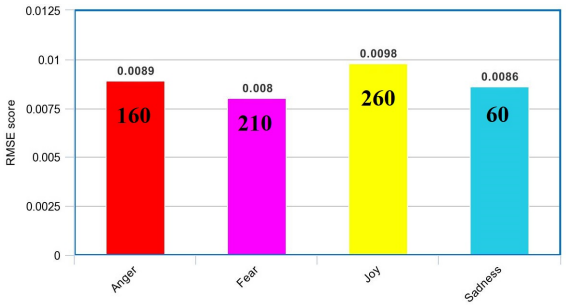

Figure 6: The best results for GBR model.

3.3.2 The best models

In Table 2 presented best models with correspondent parameters for each emotion. The best model - it's the model that give the lowest RMSE.

Emotion	Model	Parameter	RMSE
anger	SVR 'linear'	C [4]= 0.01	0.0088
fear	SVR 'linear'	C = 0.01	0.0079
joy	SVR 'poly'	C = 1.0	0.0097
sadness	SVR 'linear'	C = 0.01	0.0084

Table 2: The best models results.

The same results presented on Figure 7, with corresponded parameters in the middle of each bar plot and type of the kernel indicated in parentheses after emotions.

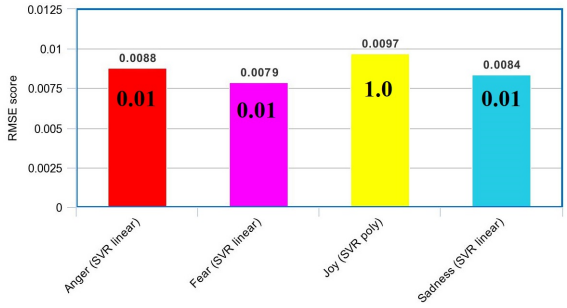

Figure 7: The best model results plot.

For anger the best model is Support Vector Regression (SVR) with linear kernel and penalty equal to 0.01.

For fear and sadness there are two best models: SVR with linear kernel and penalty equal to 0.01 and SVR with polynomial kernel, degree of polynomial function equal to 2 and penalty equal to 1.0. In the table presented only one, but their difficulties are approximately equal.

[4] 'C' for SVR means penalty parameter.

For joy there are two best model: SVR with polynomial kernel, degree of polynomial function equal to 2 and penalty equal to 1.0 and Random Forest Regressor with the number of trees equal to 160. In the table presented only SVR model, which is simpler in calculations and better.

4 Discussion

Results of this project shows that it is possible to detect emotion intensity from tweet - short text with a lot of noise, like links, emojis, hastags etc., with quite high accuracy.

However, during work occurred some difficulties. The main one was feature extracting. Mostly, features were chosen on a base of word embedding, also were attempts to use word-based features, but it is not improve results. Number of general features (such as number of tokens, average length of token etc.) was low, it can be increased, that might influence results.

Other important problem was parameters choices for different models. Models usually (especially complex ones) have a lot of parameters, and choice of their best fit (even approximately) takes large amount of time. In this work best set of parameters for all models was chosen approximately by using search for better fit among 1-2 main parameters of model.

Results could be improved in several way, one way is to perform more proper features selection, second is to try more complicated ensembles of models or neural networks.

5 Conclusion

Main idea of this project was to build models that extract emotion intensity from tweet for basic emotions: anger, fear, joy and sadness. For this purpose, datasets were filtered, features were extracted and than used as an input in different machine learning models. As result was chosen the best regression model for each emotion by RMSE.

Obtained results can be improved and used for sentiment analysis for different purposes: personal, commercial, scientific etc.

References

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. *Enhanced Sentiment Learning Using Twitter Hashtags and Smileys*, pp 241-249.

Saif M. Mohammad, and Felipe Bravo-Marquez. 2017. *Emotion Intensities in Tweets*, pp 65-77. Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (SEM 2017)

Venkatesh Duppada, and Sushant Hiray. 2017. *Tweet Emotion Intensity Estimator*. Seernet at EmoInt-2017