Toxic Message Detection

Machine learning

HSE University

2023-06-13

Primary preprocessing

- Tokenize
- Drop stop words

(file: preprocess.py)

Frequency preprocessing: Bayesian analysis

How many times is this word encounted in toxic and regular messages?

$\Longrightarrow$ how much does the given word affect the probability, that the message is toxic?

Frequency dictionary contains 150 words with the highest effect

Frequency preprocessing: formula

- T - total number of toxic messages
- t - number of times this word appears in toxic messages
- T^c and t^c are their respective complements

$$effect = \left(\frac{t^c}{T^c} - \frac{t}{T} \right) \frac{T + T^c}{t + t^c}$$

(file: preprocess_frequency.py)

RNN Preprocessing: Words

- glove.840B.300d words semantic database
- Turn each tweet into a list of vectors of length 300

(file: preprocess_rnn.py)

RNN Preprocessing: Training

- Supply encoder initial state and, sequentially, input tokens
- Obtain the final state
- Check every decoder's output against the expected input
- Update weights

The output is a 100d vector. (file: preprocess_rnn.py, rnn.py)

Fit models: Frequency Dictionary

- Input: vector of 150 floats. Each float represents how many the given message contains words from the 150 most influential words.
- Output: class 0 or 1

Fit models: RNN

- Input: vector of 100 floats, which correspond to the semantics of the message
- Output: class 0 or 1

Models fitting and comparison

Model	Train Acc	Test Acc	Train f_1	Test f_1
Freq SVM	0.9766	0.9568	0.4392	0.2447
RNN AdaBoost	0.9694	0.9614	–	–
Freq LDA	–	–	–	0.2400
RNN Random Forest	0.9863	0.6000	–	0.4440
Freq KNN	0.9450	0.9434	0.3460	0.3090
RNN Log Regression	0.9700	0.9270	–	–
Freq KNN Balanced	0.5270	0.5220	0.6590	0.6550
Freq SVM Balanced	0.6919	0.6767	0.5939	0.5765
Freq LDA Balanced	0.7930	0.7393	–	0.6710