Implicit vs. Explicit Emotion Word Recognition using Tweets

Link for dataset used in the project

Requirements

Baseline (NaiveBayes Classifier & Perceptron Classifier) - Java 8 - emoji-java-4.0.0.jar - java-json.jar

Deep Learning - Python 3 - Tensorflow - Keras - Numpy - emoji

Training

Baseline Solutions

IDE used for java : IntelliJ Idea

NaiveBayes Classifier Configuration for com.teamlab.ss18.ec.MainNaiveBayes
Program Arguments : train_data_location test_data_location

Perceptron Classifier Configuration for com.teamlab.ss18.ec.MainPerceptron
Program Arguments : train_data_location

Output will be displayed on IDE's console.

Deep Learning Solution - LSTM

change lines 407 and 408 to point to file on system

```
python3 LSTM.py
```

Deep Learning Solution - CNN

Print parameters:

```
python3 trainCNN.py --help
```

```
optional arguments:
    -h, --help            show this help message and exit --dev_sample_percentage
    Percentage of the training data to use for validation(default: 0.1)
    # Model Hyperparameters
    --embedding_dim Dimensionality of character embedding (default: 128)
    --filter_sizes Comma-separated filter sizes (default: '7,8,9')
    --num_filters, Number of filters per filter size (default: 128)
```

```
--dropout_keep_prob Dropout keep probability (default: 0.5)
--l2_reg_lambda L2 regularization lambda (default: 0.5)
--use_adam Select optimizer to use. Default is RMSPropOptimizer, else use AdamOptimizer
--activation_function Select activation function to use. Default is ReLU (default: relu)
# Training parameters
--batch_size Batch Size (default: 64)
--num_epochs Number of training epochs (default: 50)
--evaluate_every Evaluate model on dev set after this many steps (default: 50)
--checkpoint_every Save model after this many steps (default: 10)
--num_checkpoints Number of checkpoints to store (default: 1)

# Train Dataset
--traindata Traindata location

# Misc Parameters
--allow_soft_placement, True, Allow device soft device placement
--log_device_placement, False, Log placement of ops on devices
```

Train:

```
python3 trainCNN.py
```

Evaluating

```
python3 evalCNN.py --checkpoint_dir=./runs/1532529604/checkpoints/
--testdata=TEST_DATA_LOCATION
```

Replace the checkpoint dir with the output from the training. To use your own data, change the `evalCNN.py` script to load your data.

References

- Roman Klinger, Orphee de Clercq, Saif M. Mohammad, and Alexandra Balahur. 2018. Iest: Wassa2018 implicit emotions shared task. In Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Brussels, Belgium. Association for Computational Linguistics.
- Y. Kim, "Convolutional Neural Networks for sentence classification", Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1746-1751, 2014.
- Denny Britz. 2015. Implementing a CNN for Text Classification in TensorFlow. http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/.
- Denny Britz. 2015. dennybritz/cnn-text-classification- tf. https://github.com/dennybritz/cnn-text-classification-tf.

- Jason Brownlee. 2017. https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/
- https://keras.io/