Detecting depression on social media

Motivation

> With its 230 million regular users, Twitter has become such a broad stream of personal expression that researchers are beginning to use it as a tool to dig into public health problems.

- excerpt from *Time*, 2014.

The goal of this experiment is to perform sentiment analysis on random tweets and detect signs of depression in these tweets. The task is classfication of normal and depressive tweets, where depressive tweets are defined as tweets that contain depression-related keywords.

The code for this experiment is available in this notebook.

Dataset

- **Sentiment140**: the **Sentiment140** dataset containing 1,6 million tweets from the Twitter API with the 6 following attributes: *target*, *id*, *date*, *flag*, *user*, *text*. For the classification task, I took a sample of 8,000 random tweets.
- Public tweets scraped using the Twitter API: Since there is no readily available public dataset on depression, I used a Twitter scrape tool called **Twint** to collect 2,345 tweets that contain the depression-related keywords.

The two datasets are labelled respectively (`0` denotes normal tweets, `1` denotes depressive tweets) and shuffle-merged into one big dataset containing 10,345 tweets with 2 attributes: *text* and *label*.

Preprocessing

To prepare the data for training, I remove bad symbols, stop words, punctuations, and expand contractions from the tweets. The tweets are then tokenized.

The data is then split into 70% training and 30% testing.

Baseline model

The baseline model is a SVM model using TF-IDF as a weighing scheme in determining the relevance of individual words contained in the tweets. The baseline model gives a 78% accuracy score.

Training

Hyper-parameters

- Number of unique words in the vocabulary:

  `MAX_NUM_WORDS = 10,000`
- Maximum sentence length:

  `MAX_SEQ_LENGTH = 140`
- Embedding size: `EMBEDDING_DIM = 300`

Models

Naive Bayes

```
nb = Pipeline([('vect', CountVectorizer()),
               ('tfidf', TfidfTransformer()),
               ('clf', MultinomialNB()),
              ])
```

Linear Support Vector

```
sgd = Pipeline([('vect', CountVectorizer()),
                ('tfidf', TfidfTransformer()),
                ('clf', SGDClassifier(loss='hinge', penalty='l2',alpha=1e-3, random_state=42, max_iter=5, tol=Nor
               ])
```

Logistic Regression

```
logreg = Pipeline([('vect', CountVectorizer()),
                ('tfidf', TfidfTransformer()),
                ('clf', LogisticRegression(n_jobs=1, C=1e5)),
               ])
```

BiLSTM

```
inp = Input(shape=(MAX_SEQ_LENGTH,))
x = Embedding(MAX_NUM_WORDS, EMBEDDING_DIM, weights=[embedding_matrix])(inp)
x = Bidirectional(LSTM(100, return_sequences=True, dropout=0.25, recurrent_dropout=0.1))(x)
x = GlobalMaxPool1D()(x)
x = Dense(100, activation="relu")(x)
x = Dropout(0.25)(x)
x = Dense(1, activation="sigmoid")(x)
```

Results

Below are the averaged results of different models used for this task.

Models	Accuracy	F1	Precision	Recall
Baseline (SVC + TF-IDF)	78.351	68.840	61.388	78.351
Naive Bayes	91.302	90.478	92.116	91.302
Linear SVM	98.615	98.599	98.635	98.615
Logistic Regression	96.295	96.345	96.476	96.295
BiLSTM (+ word2vec)	**99.130**	**99.130**	**99.130**	**99.130**