Praktischer Teil 9

Sentiment Analyse: Mit Machine Learning

Aufgabe 1: Classifier Trainieren

In dieser Aufgabe verwenden Sie ein bestehendes Python-Programm um den Sentiment einzelner Texte mit Machine Learning zu analysieren.

- Öffnen Sie das Programm `sentimentML.py`
- Lesen Sie den Quelltext und die Dokumentation im Quelltext. Beachten Sie, dass einige Funktionen in die Datei util.py ausgelagert sind.
- Wenn Sie das Programm ausführen, werden die ersten 500 Tweets aus `dataset/SemevalTrainB.tsv` verwendet, um einen Classifier zu trainieren. Der F-Score auf dem Test-Set sollte `0.403801582691` betragen.
- Passen sie den bestehenden Code an, damit die ersten 1000 Tweets aus den Trainingsdaten verwendet wird. Verifizieren Sie, dass der F-Score `0.459778695119` beträgt.

Hinweis: Um das Programm ausführen zu können, müssten die Libraries `scipy` und `scikit-learn` via `pip` installiert worden sein.

Aufgabe 2: Lernkurve

Evaluieren Sie, wie der Finale Score von der Grösse des Trainingssets abhängt.

- Trainieren Sie dazu nacheinander auf 100, 200, 500, 1000, 2000, 4000 und allen (>8000) Texten und bestimmen Sie den F-Score.
- Stellen Sie das Ergebnis in einem Graphen dar.

Aufgabe 3: Trainingsdaten erweitern

In Aufgabe 2 haben Sie (hoffentlich) gesehen, dass der F-Score besser wird, wenn man mehr Trainingsdaten verwendet.

Darum kommt Max Schlaumeier auf die Idee, auch noch auf den Testdaten zu tranieren – er nimmt also als Traningsdaten ALLE Texte, die gegeben sind, `SemevalTrainB.tsv (tweetsTrain)` + `SemevalTestB2013.tsv (tweetsTest)`.

- Wie verändert sich der F-Score?